

الطبعة
الثالثة

سلسلة ملخصات شوم
نظريات ومسائل في

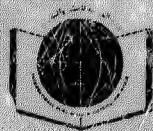
البرمجة بلغة البيسك

يتضمن قسم موسع لبيسك الحاسبات الدقيقة

بايرون س. جوتفريد

يحتوي الكتاب على ٣٨٠ مهالة محلولة
و ٥٠ مثالاً لبرامج كاملة

سلسلة ملخصات شوم في الحاسبات



الدار الدولية للنشر والتوزيع



هذا الكتاب

صدرت الطبعة الثالثة من هذا الكتاب باللغة الإنجليزية وتشمل إضافات جديدة
كان لابد من ترجمتها خدمة للقارئ العربى .

وقد تولت الدار الدولية للنشر والتوزيع مهمة طبع ونشر وتوزيع هذه الترجمة
الشاملة الجديدة انطلاقاً من رسالتها فى نشر كل جديد .

الدار الدولية للنشر والتوزيع

سبتمبر ١٩٨٩

٠١٢١٦

البرمجة بلغة البيسك

ملخصات شوم
نظريات ومساائل
ف

البرمجة بلغة البيسك

يتضمن قسم موسع لبيسك الحاسب الدقيق

تأليف

بايرون س. جوتفريد ، Ph. D.
أستاذ في الهندسة الصناعية
هندسة إدارة النظم وبحوث العمليات
جامعة بيتسبرج

مراجعة

الأستاذ الدكتور/ أحمد عزيز كمال

عميد كلية الهندسة — جامعة القاهرة
ومدير مركز بحوث الحسابات العلمية والاحصائية
جامعة القاهرة

ترجمة

ابتسام صديق أبو الخير

ماجستير حسابات علمية
نائب مدير مركز بحوث الحسابات العلمية والاحصائية
جامعة القاهرة — جمهورية مصر العربية

قام بترجمة الاضافات لهذه الطبعة الجديدة

دكتور / عباده سرحان
أستاذ مساعد بكلية الهندسة والتكنولوجيا
ومدير مركز الحساب العلمي
جامعة حلوان



الدار الدولية للنشر والتوزيع

حقوق النشر

- الطبعة الإنجليزية : حقوق التأليف © ١٩٨٦ ، ١٩٨٢ ، ١٩٧٥ دار ماكجروهيل للنشر ، إنك ، جميع الحقوق محفوظة .

Schaum's Outline of Theory and Problems of
PROGRAMMING WITH BASIC 3 / ed

Byron S. Gottfried

ISBN 0-07-023875-8

- الطبعة العربية الأولى : حقوق الطبع والنشر © ١٩٨٤ دار ماكجروهيل للنشر ، إنك — جميع الحقوق محفوظة .
- الطبعة العربية الثانية : حقوق الطبع والنشر © ١٩٨٨ ، جميع الحقوق محفوظة للناسر :
- الطبعة العربية الثالثة : حقوق الطبع والنشر © ١٩٨٩ ، جميع الحقوق محفوظة للناسر :

الدار الدولية للنشر والتوزيع

ص . ب ٥٥٩٩ هليوبوليس غرب - القاهرة

ت : ٢٥٨٢٨٨٧

تلكس : PBCRB UN ٢٠٠٧٠

فاكس : ٠٠٢٠٢ / ٢٩١٨٠٥٩

لا يجوز نشر أى جزء من هذا الكتاب أو اختزان مادته بطريقه الاسترجاع أو نقله على أى وجه أو
بأى طريقه سواء كانت اليكترونيه أو ميكانيكية أو بالتصوير أو بالتسجيل أو خلاف ذلك إلا بموافقة
الناسر على هذا كتابة ومقدماتاً .

مقدمة الناشر

المعرفة هي أصل الحضارة ،
والكلمة هي أصل المعرفة ،
والكلمة المطبوعة هي أهم مكون في هذا المصدر .

وقد كانت الكلمة المطبوعة ولا تزال أهم وسائل الثقافة والإعلام وأوسعها انتشاراً وأبقاها أثراً ، حيث حملت إلينا حضارات الأمم عبر السنين لتتولى الأجيال المتلاحقة صياغة حضارتها وإضاءة الطريق بنور العلم والمعرفة .

والكلمة تبقى مجرد فكرة لدى صاحبها حتى تتاح لها فرصة نشرها وترجمتها إلى لغات الآخرين ، ثم توزيعها ، وذلك وحده هو الذى يكفل لها أداء رسالتها .

وعالم الكتب العلمية عالم رحب ممتد الآفاق ، متسع الجنبات ، والعلم لا وطن له ولا حدود ، ويوم يحظى القارئ العربى بأحدث الكتب العلمية باللغة العربية هو اليوم الذى تتطلع له الأمة العربية جمعاء .

والدار الدولية للنشر والتوزيع تشعر بالرضا عن مساهمتها في هذا المجال بتقديم الطبوعات العربية للكتب العلمية مستهدفة توفير احتياجات القارئ العربى أستاذاً وباحثاً وممارساً .

والله ولى التوفيق ،،،

محمد وفائى كامل

المقدمة

لغة البيسك تعتبر من أسهل اللغات الشائعة حالياً في التعليم والاستخدام . وتحتوي هذه اللغة مع بساطتها الملحوظة بقوة كافية واستمالات متعددة يمكن أن يستفيد منها العديد من الناس المختلفين ولأنواع عديدة من التطبيقات . وتبعاً لذلك فإن دروس لغة البيسك BASIC أصبحت شائعة في عديد من المدارس الثانوية وكذلك المدارس الإعدادية بالإضافة إلى معظم الكليات والجامعات . والآن المدارس الأولية بالولايات المتحدة الأمريكية تعرض مقدمة لدروس البيسك لمجموعات متنوعة من الطلبة .

سبب آخر لشعبية البيسك هو انتشاره الواسع ؛ فاللغة الآن متاحة عموماً على كل الحاسبات الكبيرة ، ومدعمة فعلياً بواسطة كل الخدمات للمشاركة الزمنية التجارية . وعلاوة على ذلك ، فإن البيسك أصبح اللغة الأساسية لمعظم تطبيقات الحاسبات الدقيقة . ولذلك فإن اللغة يمكن استخدامها على كل الحاسبات الكبيرة والصغيرة وفي كل أنواع بيئات البرمجة المختلفة .

ويعرض هذا الكتاب تعليمات في برمجة الحاسب باستخدام الخصائص القياسية للغة البيسك وقد تمت مناقشة كل الخصائص الأساسية للغة البيسك . وبالإضافة إلى ذلك ، فإن الكتاب يبرز التطور في البرامج من حيث المنطق والكفاءة والنظام . ولذلك فإن القارئ يتعرض لأساسيات التدريب على البرمجة الجيدة بجانب القواعد المحددة في لغة البيسك .

طريقة الكتابة بطريقة أولية وبصورة متعمدة . وجعلت الكتاب سهل الفهم بواسطة قاعدة عريضة من القراء المستمعين يتراوح بين طلبة المدارس الثانوية والموظفين . ويناسب هذا الكتاب مستوى المدارس الثانوية المتقدمة أو المستوى الجامعي للطلبة المبتدئين ككتاب أساسي في محاضرات البرمجة وكنص مكمّل لمحاضرات أكثر شمولاً في الطرق الفنية التحليلية أو كمرشد فعال للدراسة بدون معلم . وفي معظم أجزائه مستوى الرياضيات المطلوبة لا تتجاوز جبر المدارس العليا .

وفي هذه الطبعة الثالثة تم التوسع كثيراً في المادة الخاصة ببيسك الحاسب الدقيق ، وتم تنظيم الكتاب في ثلاثة أجزاء رئيسية هي : البيسك الأول ، البيسك المتقدم ، وبيسك الحاسب الدقيق . ويمثل الجزء الأول — خصائص اللغة شائعة الاستخدام . ويمكن تدريس مقرر مختصر في البرمجة في هذه المادة وحدها . ويتعلق الجزء الثاني بخصائص أكثر تخصصاً مثل البرامج وجمل المصفوفات ، والتعامل مع الملف ؛ ويركز الجزء الثالث على التحسينات المتاحة في سبيل الحاسب الدقيق ، مع تأكيد خاص على ميكروسوفت بيسك ، كما ينفذ على حاسب IBM الشخصي . وتحتوي هذه المادة على فصل مختصر في البرمجة سهلة الاستخدام ، وفصل أطول على بيانات الحاسب الدقيق .

تقدم المادة بطريقة تمكن القارئ من كتابته برامج بيسك كاملة ولكن بسيطة بأسرع ما يمكن . ومن المهم أن يكتب القارئ مثل هذه البرامج ويشغلها على الحاسب في نفس الوقت الذي يدرس فيه الكتاب . وهذا يزيد من ثقة المبرمج الحديث في نفسه وتثير من حماسه لهذا الموضوع (تعلم كيفية برمجة الحاسب كتعلم كيفية العزف على آلة موسيقية ، لا يمكن تعلمها بسهولة بدراسة كتاب !) .

يتضمن هذا الكتاب عدداً كبيراً من الأمثلة كجزء مكمّل له . وهذه تتضمن عدداً من مسائل البرمجة الشاملة بجانب أنواع التدريبات العادية . بالإضافة إلى ذلك ، تتضمن نهاية معظم الفصول مجموعة من المسائل المحلولة . ويجب دراسة هذه الأمثلة والمسائل المحلولة بكل عناية أثناء تقدم القارئ في دراسته لفصول الكتاب وتدريبه على كتابة البرامج الخاصة به .

وتتضمن أيضاً مجموعات من أسئلة المراجعة والمسائل التكميلية ومسائل البرمجة في نهاية كل فصل . تمكن أسئلة المراجعة القارئ من اختبار استرجاعه للمواد المقدمة في الفصل . وتمتد أيضاً بملخص مؤثر للفصل . ولا تتطلب معظم المسائل التكميلية ومسائل البرمجة أي خلفية خاصة في الرياضيات أو التكنولوجيا . ويجب على الطالب أن يحل هذه المسائل بقدر المستطاع (الإجابة على هذه المسائل التكميلية أيضاً متاحة في نهاية الكتاب) . وعند استخدام هذا الكتاب كنص محاضرات ، فإننا ننصح القائم بالتدريس بأن يكمل مسائل البرمجة ببرامج إضافية تتلام مع اهتمامات الطالب الخاصة .

الخصائص الأساسية المهمة للغة ملخصة داخل الغلاف الخلفى وفي خمسة ملاحق وذلك لتسهيل مهمة القارئ . يجب أن تستخدم هذه المادة باستمرار كمرجع جاهز واسترجاع سريع . وسوف تكون مفيدة عند كتابة أو متابعة البرنامج الجديد .

وأخيراً ، فإن القارئ الذى يكل هذا الكتاب سيكون قد تعلم الكثير عن مفاهيم لغات الحاسب العامة بجانب القواعد الخاصة بلغة لبيسك ويجب أن يكون مقتنئاً تماماً أن البرمجة بلغة اليبسك ليست سهلة فقط ولكنها مسلية .

بأبرون س . جوفريد

المحتويات

صفحة

٧	مقدمة الناشر
٩	مقدمة الكتاب
١١	أمثلة لمجموعة برامج كاملة

الجزء الأول : أساسيات لغة البيسك

الفصل ١ : مفاهيم تمهيدية

١٧	١ — مقدمة للحاسبات
١٧	١ — ٢ خصائص الحاسب
٢٠	١ — ٣ أساليب التشغيل
٢٤	١ — ٤ أنواع لغات البرمجة
٢٥	١ — ٥ مقدمة للغة البيسك

الفصل ٢ : طريقة البداية للغة البيسك

٣٣	٢ — ١ الأرقام « الثوابت »
٣٣	٢ — ٢ سلاسل الحروف
٣٤	٢ — ٣ المتغيرات
٣٤	٢ — ٤ المعاملات والصيغ الرياضية « التعبيرات الرياضية »
٣٥	٢ — ٥ التدرج الهرمي للعمليات الحسابية
٣٦	٢ — ٦ استخدام الأقواس
٣٦	٢ — ٧ قواعد خاصة متعلقة بالصيغ الرياضية
٣٧	٢ — ٨ تحديد قيم جملة LET
٣٩	٢ — ٩ قراءة المدخلات — جملة INPUT
٤٠	٢ — ١٠ طباعة المخرجات — جملة PRINT
٤٤	٢ — ١١ جملة END
٤٥	٢ — ١٢ كتابة برامج كاملة بلغة البيسك
٤٦	٢ — ١٣ التعليقات على البرنامج — جملة REM
٤٧	٢ — ١٤ تحويل التحكم — جملة GO-TO
٤٧	٢ — ١٥ تكرار تنفيذ البرنامج
٤٨	٢ — ١٦ ملاحظات ختامية

صفحة

٦٣	الفصل ٣ : تشغيل برنامج بيسك
٦٣	٣ - ١ النهاية الطرفية للمشاركة الزمنية
٦٤	٣ - ٢ التسجيل لدخول النظام
٦٦	٣ - ٣ ادخال البرنامج
٦٧	٣ - ٤ تصحيح الأخطاء
٦٨	٣ - ٥ تشغيل برنامج
٧١	٣ - ٦ التسجيل للخروج من النظام
٧٢	٣ - ٧ تحليل الأخطاء
٧٣	٣ - ٨ تحديد الأخطاء المنطقية
٧٦	٣ - ٩ ملاحظات ختامية
٨١	الفصل ٤ : التفرع وتكوين حلقات تكرارية
٨١	٤ - ١ المعاملات الرابطة
٨٢	٤ - ٢ التفرع المشروط — جملة IF THEN
٨٧	٤ - ٣ التفرع المتعدد — جملة ON- GOTO
٨٨	٤ - ٤ جملة STOP
٩٣	٤ - ٥ تكوين الحلقات التكرارية FOR TO
٩٤	٤ - ٦ انتهاء حلقة تكرارية — جملة NEXT
٩٨	٤ - ٧ الحلقات التكرارية المتداخلة
١١٥	الفصل ٥ : بعض الخصائص الإضافية للغة البيسك
١١٥	٥ - ١ الدوال المكتوبة
١١٩	٥ - ٢ القوائم والجداول « المجموعات المتراسة »
١١٩	٥ - ٣ المتغيرات ذات الأدلة
١٢٣	٥ - ٤ تعريف المجموعات المتراسة — جملة DIM
١٢٤	٥ - ٥ إدخال بيانات الإدخال — جملتي DATA, READ
١٣٣	٥ - ٦ إعادة قراءة البيانات — جملة RESTORE
١٣٤	٥ - ٧ ملاحظات ختامية

الجزء الثاني : البيسك المتقدم

١٥١	الفصل ٦ : الدوال والبرامج الفرعية الصغيرة
١٥١	٦ - ١ تعريف الدالة — جملة DEF
١٥١	٦ - ٢ الإشارة إلى الدالة
١٥٧	٦ - ٣ الدوال المتعددة السطور
١٦٠	٦ - ٤ توكيد وفك شفرة البيانات — جملة CHANGE

ملحة

١٦٢.....	٦ — ٥ دالتى CHR\$, ASC.....
١٦٧.....	٦ — ٦ توليد أرقام عشوائية — دالة RND.....
١٦٨.....	٦ — ٧ جملة RANDOMIZE.....
١٧٢.....	٦ — ٨ تعريف برنامج فرعى.....
١٧٣.....	٦ — ٩ الإشارة إلى برنامج فرعى — جملة GOSUB.....
١٨٢.....	٦ — ١٠ المخرجات البيانية.....
٢٠٣.....	الفصل ٧ : المتجهات والمصفوفات.....
٢٠٣.....	٧ — ١ عمليات المتجهات والمصفوفات.....
٢٠٧.....	٧ — ٢ إدخال / إخراج المتجهات والمصفوفات.....
٢١٦.....	٧ — ٣ مصفوفات خاصه.....
٢٢٤.....	٧ — ٤ تغيير أبعاد المجموعات المتراصة.....
٢٤٣.....	الفصل ٨ : ملفات البيانات.....
٢٤٣.....	٨ — ١ ملفات البيانات التسلسليه.....
٢٥٢.....	٨ — ٢ ملفات البيانات العشوائية.....
٢٦١.....	٨ — ٣ مواصفات ملف أثناء وقت التشغيل.....

الجزء الثالث : بيسك الحاسب الدقيق

٢٧٢.....	الفصل ٩ : التحسينات على البيسك.....
٢٧٣.....	٩ — ١ توسعات أولية فى اللغة.....
٢٨٠.....	٩ — ٢ جمل إضافية.....
٢٨٩.....	٩ — ٣ دوال مكتبية إضافية.....
٢٩٦.....	٩ — ٤ ملفات البيانات للحاسبات الدقيقة.....
٣٠١.....	٩ — ٥ إجراءات لغة الآلة فى البيسك.....
٣٠٢.....	٩ — ٦ خواص مستبعده عن بيسك الحاسبات الدقيقة.....
٣٠٩.....	الفصل ١٠ : بيئة الحاسب الدقيق.....
٣٠٩.....	١٠ — ١ تمييز الملفات.....
٣١١.....	١٠ — ٢ أوامر نظام الحاسب الدقيق.....
٣١٤.....	١٠ — ٣ شاشة عرض TV.....
٣١٧.....	١٠ — ٤ لوحة المفاتيح.....
٣١٩.....	١٠ — ٥ وحدات إدخال أخرى قابلة للبرمجة.....
٣٢٥.....	١٠ — ٦ استخدام اللون والصوت.....
٣٣٠.....	١٠ — ٧ تنقيح البرنامج.....

صفحة

٣٣٧	الفصل ١١ : البرمجة سهلة الاستخدام
٣٣٧	١١ - ١ التلقينات
٣٣٩	١١ - ٢ القوائم
٣٤٩	١١ - ٣ فحص الخطأ
٣٥٣	١١ - ٤ تحقق المستخدم
٣٥٩	الفصل ١٢ : بيانات الحاسب الدقيق
٣٥٩	١٢ - ١ أساسيات البيانات
٣٦١	١٢ - ٢ النقط والسطور
٣٧٢	١٢ - ٣ الأشكال
٣٨٤	١٢ - ٤ الرسوم المتحركة
٣٩١	١٢ - ٥ بيانات الحروف
٤٠٣	اجابات لمسائل تكميلية مختارة
٤١٩	قائمة المصطلحات العلمية (عربى - انجليزى)
٤٢٧	الفهرس الأبنجدى :
٤٣٣	ملحق (أ)
٤٣٥	ملحق (ب)
٤٣٦	ملحق (ج)
٤٣٧	ملحق (د)
٤٤٥	ملحق (هـ)

أمثلة لمجموعة برامج كاملة :

- (١) جذور معادلة تربيعية (أمثلة ٢ — ٢٦ ، ٢ — ٣٠) — ٤٥ ، ٤٧
- (٢) مساحة ومحيط دائرة (مثال ٣ — ٨ ، ٣ — ٩) — ٧٠
- (٣) جذور معادلة جبرية (مثال ٤ — ٥) — ٨٣
- (٤) حساب قيمة الاستهلاك (مثال ٤ — ٩) — ٨٨
- (٥) إيجاد متوسط بيانات تلوث الهواء (مثال ٤ — ١٦) — ٩٥
- (٦) توليد أرقام فيبوناتشي والبحث عن الأرقام الأولية (مثال ٤ — ١٨) — ١٠٠
- (٧) جدول للدوال (مثال ٥ — ٥) — ١١٧
- (٨) كلمة غير مرتبة (مثال ٥ — ٩) — ١٢٠
- (٩) إعادة ترتيب قائمة من الأرقام (مثال ٥ — ١٤) — ١٢٦
- (١٠) معالجة عناصر الجدول (مثال ٥ — ١٥) — ١٢٩
- (١١) البحث عن أكبر قيمة (مثال ٦ — ٦) — ١٥٣
- (١٢) مولد بجلاتين (مثال ٦ — ١٥) — ١٦٣
- (١٣) لعبة حظ (اصطياد كرايس) (مثال ٦ — ٢٠) — ١٦٨
- (١٤) برنامج المراتب الشهرية (مثال ٦ — ٢٦) — ١٧٦
- (١٥) محاكاة لإرتداد كره (مثال ٦ — ٢٨) — ١٨٤
- (١٦) التعامل مع مصفوفة (مثال ٧ — ١٥) — ٢١٤
- (١٧) معكوس المصفوفة (مثال ٧ — ١٨) — ٢٢٠
- (١٨) المعادلات الآتية (مثال ٧ — ١٩) — ٢٢٢
- (١٩) توفير منحني المربعات الصغرى (لعبة الاسواق المالية) (مثال ٧ — ٢٢) — ٢٢٥
- (٢٠) تشغيل درجات اختبار طالب (مثال ٨ — ٤) — ٢٤٧
- (٢١) مراتبة المخزون (مثال ٨ — ١٠) — ٢٥٨
- (٢٢) البحث في ملف بيانات (مثال ٨ — ١٣) — ٢٦١
- (٢٣) توليد أرقام فيبوناتشي والبحث عن الأرقام الأولية وتشغيلها على حاسب دقيق (مثال ٩ — ١١) — ٢٧٩
- (٢٤) البحث عن القيمة العظمى على حاسب دقيق (مثال ٩ — ٢٧) — ٢٨٨
- (٢٥) توليد بجلاتين على حاسب دقيق (مثال ٩ — ٢٨) — ٢٩٢
- (٢٦) خلق ملف بيانات متتالي (مثال ٩ — ٢٩) — ٢٩٧
- (٢٧) تشغيل درجات اختبار طالب على حاسب دقيق (مثال ٩ — ٣٠ ، ١٠ — ٢٠) — ٣٣١ ، ٣٠٠
- (٢٨) التحكم في المخزون على حاسب دقيق (مثال ٩ — ٣١) — ٣٠٠
- (٢٩) زمن اليوم (مثال ١٠ — ٥) — ٣١٢
- (٣٠) برجة عرض على شاشة تليفزيونية (لا شيء يمكن أن يصبح خطأ .. خطأ .. خطأ ..) (مثال ١٠ — ١٠ ، ١٠ — ١٧) — ٣١٦ ،

٣٢٨

- (٣١) برجة مفاتيح الدوال (مثال ١٠ — ١١) — ٣١٨
- (٣٢) برجة قلم ضوئي (مثال ١٠ — ١٢) — ٣٢٠
- (٣٣) برجة عصا توجيه (مثال ١٠ — ١٣) — ٣٢٣
- (٣٤) النص متعدد الألوان (مثال ١٠ — ١٤) — ٣٢٥
- (٣٥) برجة البوق (صفارة انذار) (مثال ١٠ — ١٦) — ٣٢٧
- (٣٦) ادخال درجات اختبار طالب (مثال ١١ — ٢ ، ١١ — ٦) — ٣٣٨ ، ٣٥٠
- (٣٧) التحويل الشخصي (حسابات الربح المربك) (مثال ١١ — ٤) — ٣٤١
- (٣٨) تخزين بيانات المعامل (مثال ١١ — ٧) — ٣٥٣
- (٣٩) نقط في الفراغ (مثال ١٢ — ٨) — ٣٦٣

- (٤٠) السهم المضيء (مثال ١٢ - ١٠ ، ١٢ - ١٩ - ٣٦٤ ، ٣٧٧ .
- (٤١) الخطوط المتحركة (الفن الحركي) (مثال ١٢ - ١١) - ٣٦٥
- (٤٢) الانحدار الخطي مع عرض بياني (مثال ١٢ - ١٢) - ٣٦٨ .
- (٤٣) المستطيلات المتعددة (مثال ١٢ - ١٤) - ٣٧٣ .
- (٤٤) المشكّال (مثال ١٢ - ١٦) - ٣٧٤ .
- (٤٥) الدوائر المتعددة (مثال ١٢ - ١٨) - ٣٧٦ .
- (٤٦) مولد خريطة دائرية (مثال ١٢ - ٢٤) - ٣٨٠
- (٤٧) منطاد بنص متحرك (مثال ١٢ - ٢٧) - ٣٨٣
- (٤٨) محاكاة كرة مرتدة على الحاسب الدقيق (مثال ١٢ - ٢٨ ، ١٢ - ٢٩) - ٣٨٥ ، ٣٨٧
- (٤٩) لعبة كرة التجديف (مثال ١٢ - ٣٠) - ٣٨٨
- (٥٠) مولد خريطة الأعمدة (مثال ١٢ - ٣١ ، ١٢ - ٣٢) - ٣٩١ ، ٣٩٣

الجزء الأول : أساسيات لغة البيسك

الفصل ١

مفاهيم تمهيدية .

Introductory Concepts

يعرض هذا الكتاب تعليمات لبرمجة الحاسب باستخدام لغة برمجة سهلة في تعلمها وشائعة تسمى بيسك (Beginner's All-purpose Symbolic (BASIC Instruction Code) وسوف نرى كيف يمكن تحليل مسألة وصفت مبدئياً بالكلمات ونحولت أخيراً إلى برنامج بيسك قابل للتنفيذ . وقد عرضت هذه المفاهيم بصورة توضيحية ، بواسطة العديد من المسائل المختارة .

١ - ١ مقدمة للحاسبات INTRODUCTION TO COMPUTERS

توجد الحاسبات حالياً في أشكال وأحجام وأسعار مختلفة وتقوم العديد من المؤسسات الكبيرة والجامعات والمستشفيات والمصالح الحكومية باستخدام الحاسبات الضخمة ذات الأغراض العامة للقيام بالحسابات التجارية والعلمية المعقدة . وتعرف هذه بالحاسبات الكبيرة وهي غالية الثمن جداً (بعضها يبلغ ثمنه ملايين الدولارات) وتحتاج إلى بيئة يمكن التحكم فيها بعناية (درجة الحرارة والرطوبة ... الخ) وكقاعدة عامة فإنه من غير المسموح لمن يستخدموها التعامل معها مباشرة .

وهذه الحاسبات الكبيرة أصبحت متاحة منذ أوائل الخمسينيات رغم إنه لم يكن هناك إلا عدد قليل من الأفراد يعلمون كيفية استعمالها في السنوات الأولى . هؤلاء الأفراد والموظفين الذين استخدموها كانوا عموماً من العلماء والمهندسين والمحاسبين وعلى درجة عالية من التدريب ، وعلى ذلك فإنه لم يكن غريباً أن ينظر عامة الناس للحاسبات بنظرة الشك والرهبة . وفي خلال الستينيات أصبح من المعتاد أن يقوم الطلاب في الجامعات بتعلم كيفية برمجة الحاسبات الكبيرة (عادة باستخدام لغة Basic أو أى لغة برمجة أخرى من المستوى الرفيع) ونتيجة لذلك فقد بدأ يختفى بعض الغموض المرتبط باستخدام الحاسبات .

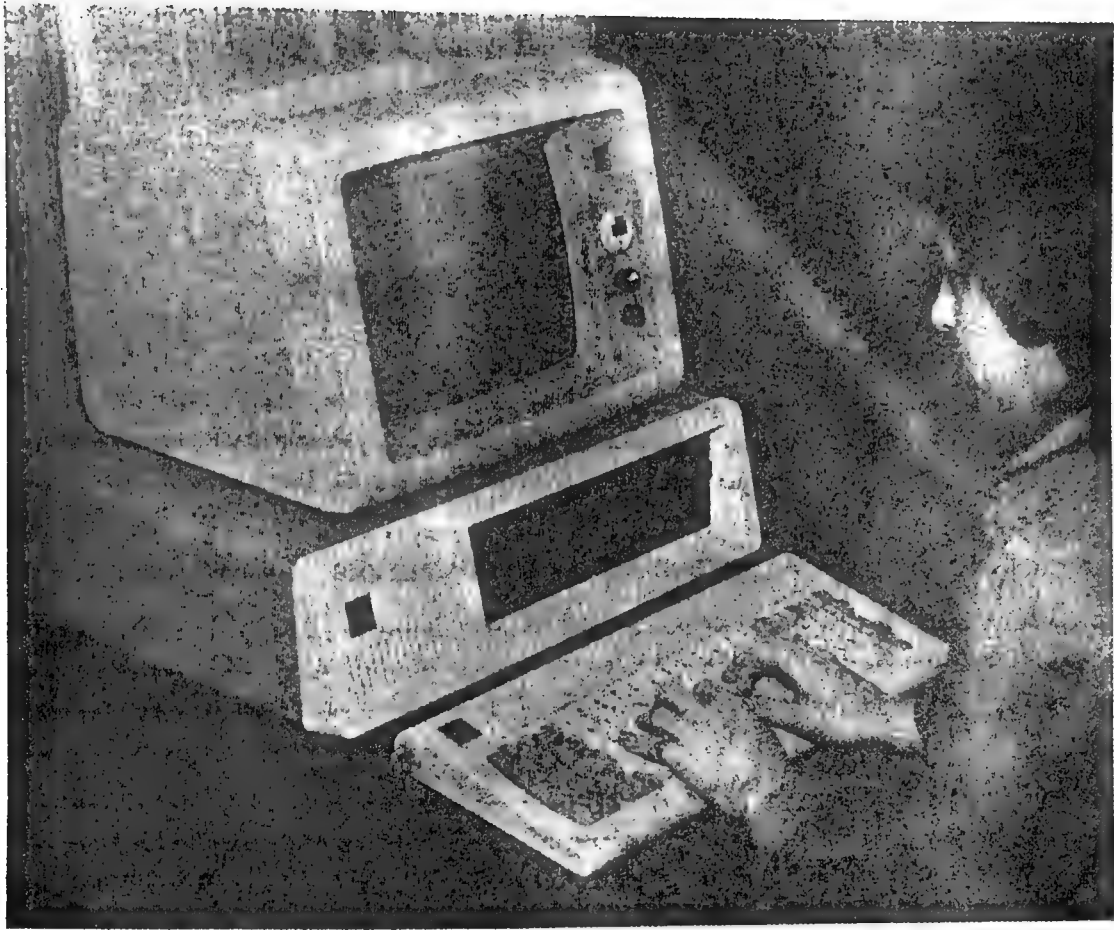
وقد شاهدت أواخر الستينيات وأوائل السبعينيات ظهور حاسبات صغيرة أقل تكلفة وأصغر حجماً وهذه الأجهزة تعطي كفاءة الحاسبات الكبيرة السابقة بجزء من تكلفتها فقط ، مما أمكن معه للعديد من المدارس والمؤسسات التجارية الحصول على حاسبات صغيرة عندما أصبحت متاحة ، والتي لم يكن في إمكانها الحصول على حاسبات كبيرة . وقد نتج عن التقدم في تكنولوجيا الدوائر المتكاملة (شرائح السليكون) الوصول إلى حاسبات أصغر كثيراً وأقل تكلفة تسمى الحاسبات الدقيقة ، ولما كانت هذه الأجهزة قد بنيت بالكامل من الدوائر المتكاملة ولذا لم تكن أكبر كثيراً أو أغلى كثيراً من آلة كاتبة عادية .

وعادة ما يشار إلى الحاسبات الدقيقة على إنها حاسبات شخصية حيث أن معظمها يستخدم بواسطة شخص واحد في لحظة واحدة ، وشكل (١-١) يبين طالب يستخدم حاسب شخصي .

وهناك العديد من الحاسبات الدقيقة الحديثة تقترب قوتها الحاسوبية من الحاسبات الصغيرة ، وتستمر في تحسين كفاءتها بشكل كبير في الوقت الذي تقل أسعارها باستمرار ، وعلى ذلك فإنه يمكن استخدامها في العديد من التطبيقات الفنية والتجارية والتعليمية والشخصية ولذلك فإننا نجد أن الحاسبات الدقيقة منتشرة بكثرة في العديد من المدارس والمؤسسات التجارية ، وعلى ما يبدو فإننا سنتنشر قريباً كأحد الوحدات المنزلية الشائعة ، وهناك العديد من المؤسسات الكبيرة التي تستخدم الحاسبات الدقيقة كنهائيات طرفية أو محطات عمل ويتم توصيلها بحاسبات أكبر أو حاسبات دقيقة أخرى عن طريق شبكة اتصالات ، وعندما تستخدم بهذه الطريقة فإن الحاسبات الدقيقة تميل إلى اتمام الاستخدامات على الحاسبات الكبيرة بدلاً من أن تحمل محلها .

١ - ٢ خصائص الحاسب COMPUTER CHARACTERISTICS

ومهما تكن أحجام الحاسبات الرقمية ، فهي تتكون أساساً من وحدات إلكترونية يمكنها إرسال وتخزين وتداول المعلومات (أى البيانات) . ويوجد نوعان مختلفان من البيانات وهما : بيانات رقمية وبيانات هجائية (مثل الأسماء والعناوين .. الخ) .



شكل ١ - ١

تتطلب التطبيقات العلمية والفنية أساساً تشغيل بيانات رقمية ، بينما تتضمن التطبيقات التجارية عادة تشغيل كل من البيانات الهجائية والرقمية . وتستخدم بعض الحاسبات فقط لتشغيل البيانات الهجائية التي ترد في النصوص المكتوبة (مثل الخطابات ومخطوطات الكتب .. إلخ) ، وتعرف هذه بمعالجة الكلمات .

ومن أجل تشغيل مجموعة معينة من البيانات فيجب أن يعطى الحاسب مجموعة صحيحة من التعليمات تسمى برنامج . ويتم ادخال هذه التعليمات في الحاسب ثم يتم تخزينها في جزء من ذاكرة الحاسب .

وفي أى وقت يتم فيه تنفيذ برنامج سبق تخزينه يحدث ما يلي :

١ - يتم إدخال مجموعة من المعلومات ، تسمى المدخلات (من خلال نهاية طرفية مثل الآلة الكاتبة أو من خلال قارئ البطاقات أو .. إلخ) ويتم تخزينها في جزء آخر من ذاكرة الحاسب .

٢ - بعد ذلك يتم تشغيل هذه المدخلات لإصدار نتائج معينة مطلوبة تسمى بالمخرجات .

٣ - ثم تطبع المخرجات (وربما جزء من المدخلات) على فرغ من الورق أو تعرض على شاشة مرئية TV .

ويمكن تكرار هذا الإجراء ذى الخطوات الثلاث عدة مرات إذا تطلب الأمر ذلك ، وبذا يتم تشغيل كمية كبيرة من البيانات في توال سريع . ويجب أن يكون مفهوماً أن كل هذه الخطوات وبالذات الخطوة رقم ٢ ورقم ٣ يمكن أن تكون طويلة ومقدة .

مثال ١ - ١

تم كتابة برنامج للحاسب لحساب مساحة دائرة باستخدام الصيغة الرياضية $A = \pi r^2$ عند إعطاء قيمة رقمية لنصف القطر r كدخول ، وسوف يتضمن التنفيذ الخطوات التالية :

١ - قراءة القيمة الرقمية لنصف قطر الدائرة .

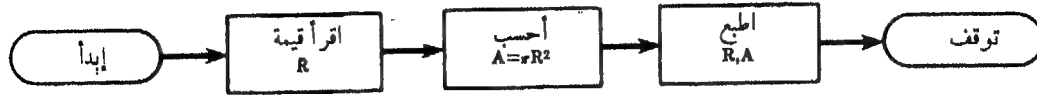
٢ - حساب قيمة المساحة باستخدام الصيغة الرياضية السابقة (وهذه القيمة سيتم تخزينها مع المدخلات في ذاكرة الحاسب) .

٣ - طباعة (عرض) قيمة نصف القطر وقيمة المساحة المناظرة .

٤ - إيقاف التشغيل .

وكل خطوة من هذه الخطوات سنحتاج إلى تعليمة واحدة أو أكثر في برنامج الحاسب .

يمكن تمثيل جميع الإجراءات بطريقة تصويرية كالمبينة في شكل ١ - ٢ . وتعرف بخريطة سير العمليات . تساعد خرائط سير العمليات القارئ في تصور طريقة انسياب المنطق في البرنامج .



شكل ١ - ٢

توضح المناقشة السابقة خاصيتين مهمتين من خواص الحاسب الرقمي وهما : الذاكرة والمقدرة على البرمجة . وخاصية ثالثة أخرى مهمة ألا وهي السرعة والاعتمادية . وسوف نذكر الكثير عن الذاكرة والسرعة والاعتمادية في الفقرات القليلة القادمة . أما موضوع البرمجة فسوف يتم مناقشته بالتفصيل خلال الأجزاء الباقية من هذا الكتاب .

الذاكرة Memory

يكون كل جزء من المعلومات التي تخزن في ذاكرة الحاسب كتوافقية فريدة من الأرقام صفر وواحد تسمى هذه الأرقام من صفر وواحد أرقاماً ثنائية *bits* (binary digits) . يمثل كل رقم ثنائي بواسطة وحدة إلكترونية وهي إما مغلقة (صفر) أو « مضاءة » (واحد) .

معظم الحاسبات الصغيرة لها ذاكرة منظمة على أساس مضاعفات ثمانية أرقام ثنائية تسمى *bytes* (بايتات) . وعادة يمثل كل حرف بايت واحد (حرف هجائي أو رقم مفرد أو رمز لعلامات الوصل) ويمكن أن يحتل الأمر بايتاً أو اثنين أو ثلاثة ، ويمكن أن تحتل الكمية الرقمية أى مكان من 2 إلى 8 بايت (معتمداً في ذلك على نوع الرقم ومدى دقته) .

ويمكن عادة التعبير عن حجم ذاكرة الحاسب ببعض مضاعفات 2^{10} أى 1024 بايت . ويشار إليها (IK) . وتتراوح أحجام ذاكرة الحاسبات الصغيرة من 64 K إلى 1024 K بايت (IMB)

مثال ١ - ٢

سعة ذاكرة حاسب شخصي 256 K بايت وعلى ذلك يمكن تخزين 256×1024 يساوي 262,144 حرف أو تعليمه في ذاكرة الحاسب . إذا استخدمت الذاكرة كلها لتمثل بيانات حرفية فيمكن تخزين حوالى 3200 اسم وعنوان بداخل الحاسب في وقت واحد (وذلك بفرض 80 حرفاً لكل اسم وعنوان) .

أما إذا استخدمت الذاكرة لتمثيل بيانات رقمية وليست أسماء وعناوين فيمكن تخزين حوالى 65,000 كمية في وقت واحد (بفرض 4 بايت لكل رقم) .

وتنظم ذاكرة الحاسبات الكبيرة في صورة كلمات *words* وليست بايتات . وتتكون كل كلمة من عدد كبير نسبياً من الأرقام الثنائية ، والرقم النموذجي يتراوح ما بين 32 و 36 وذلك يسمح لكيفية رقمية أو مجموعة صغيرة من الحروف (كنموذج 4 أو 5) أن تمثل بداخل كلمة واحدة في الذاكرة . ويعبر عادة عن ذاكرة الحاسبات الكبيرة كضاعفات (1 K) أى $(2^{10} = 1024)$ كلمة . ويكون للحاسبات الكبيرة ذاكرة بها عدة ملايين من الكلمات .

مثال ١ - ٣

ذاكرة حاسب كبير متعددة الأغراض لها سعة 2048 K والتي تساوي 2048×1024 أى 2,097,152 كلمة . إذا استخدمت الذاكرة كلها لتمثيل بيانات رقمية فيمكن تخزين أكثر من 2 مليون رقم بداخل الحاسب في وقت واحد .

أما إذا استخدمت الذاكرة لتمثيل حروف بدلا من بيانات رقمية فيمكن تخزين حوالى 8 ملايين حرف في وقت واحد . وتكون هذه الذاكرة كافية لتخزين محتويات كتاب بالكامل .

وتستخدم معظم الحاسبات وحدات ذاكرة مساعدة (من أمثلة ذلك شرائط مغناطيسية وأقراص وحدات ذاكرة صلبة) بالإضافة إلى ذاكرتها الأساسية . وتتراوح هذه الوحدات من بضع مئات الآلاف من البايت (لحاسب صغير) إلى عدة ملايين من الكلمات (لحاسب كبير) . وبالإضافة إلى ذلك فإن هذه الوحدات تسمح بتسجيل المعلومات تسجيلاً دائماً ، حيث يمكن لها أن توضع على الحاسب أو تؤخذ وتخزن في مكان آخر في حالة عدم استخدامها . وبذلك فإن وقت التوصل (أى الوقت اللازم لتخزين أو استرجاع المعلومات) يكون أكثر بكثير لهذه الوحدات عنها للذاكرة الأساسية .

السرعة والاعتمادية Speed and Reliability

ولأن سرعة الحاسب فائقة فيمكنه أن يقوم بعمليات حسابية في بضع دقائق يلزمها شهور أو ربما سنين إذا حسبت باليد . ويمكنه القيام بالمهام البسيطة مثل جمع رقمين في جزء صغير من ميكروثانية ($1 \mu s = 10^{-6} s$) . من الناحية العملية يمكن حساب درجات نهاية الفصل الدراسي لكل الطلاب الموجودين في جامعة كبيرة في بضع دقائق من وقت الحاسب .

ويصاحب هذه السرعة الفائقة مستوى عالٍ مساو لها في درجة الاعتمادية . ذلك أن الحاسب لا يرتكب عملياً أخطاء بنفسه . فأخطاء الحاسب المعلن عنها ، مثل تسلم شخص فاتورة شهرية بأكثر من مليون دولار من إحدى المحلات المحلية ، تكون كلها تقريباً نتيجة أخطاء برمجية أو خطأ في إرسال البيانات .

١ - ٣ أساليب التشغيل 1.3 MODES OF OPERATION

يمكن الانقطاع بإمكانية الحاسب الرقبي بطريقتين مختلفتين . هاتان الطريقتان هما أسلوب التشغيل على دفعات وأسلوب التشغيل التحويلي وكلاهما شائع . ولكل مزايان لأنواع معينة من المسائل .

التشغيل على دفعات Batch Processing

في الأيام الأولى للعمليات الحسابية كان يتم التشغيل عن طريق التشغيل على دفعات ، ومازالت هذه الطريقة تستخدم في بعض المدارس والمؤسسات التجارية إلا أنها أقل شيوعاً عما كانت عليه سابقاً .

في التشغيل على دفعات يتم قراءة عدد من الشغلات في الحاسب وتخزن داخلياً ثم يتم تشغيلها على التوالي . (تشير الشغلة إلى برنامج الحاسب ومجموعة البيانات الملازمة له والمراد تشغيلها) وعادة يسجل البرنامج والبيانات على بطاقات مثقبة . تقرأ المعلومات المسجلة على بطاقات للحاسب بواسطة قارئ بطاقات ميكانيكى ، ثم يتم تشغيلها وبعد انتهاء تشغيل الشغلة ، تطبع النتائج مع قائمة للبرنامج على أفرخ كبيرة من الورق بواسطة آلة طباعة ذات سرعة عالية ، وهذا النوع من التشغيل على دفعات أصبح طراز قديم .

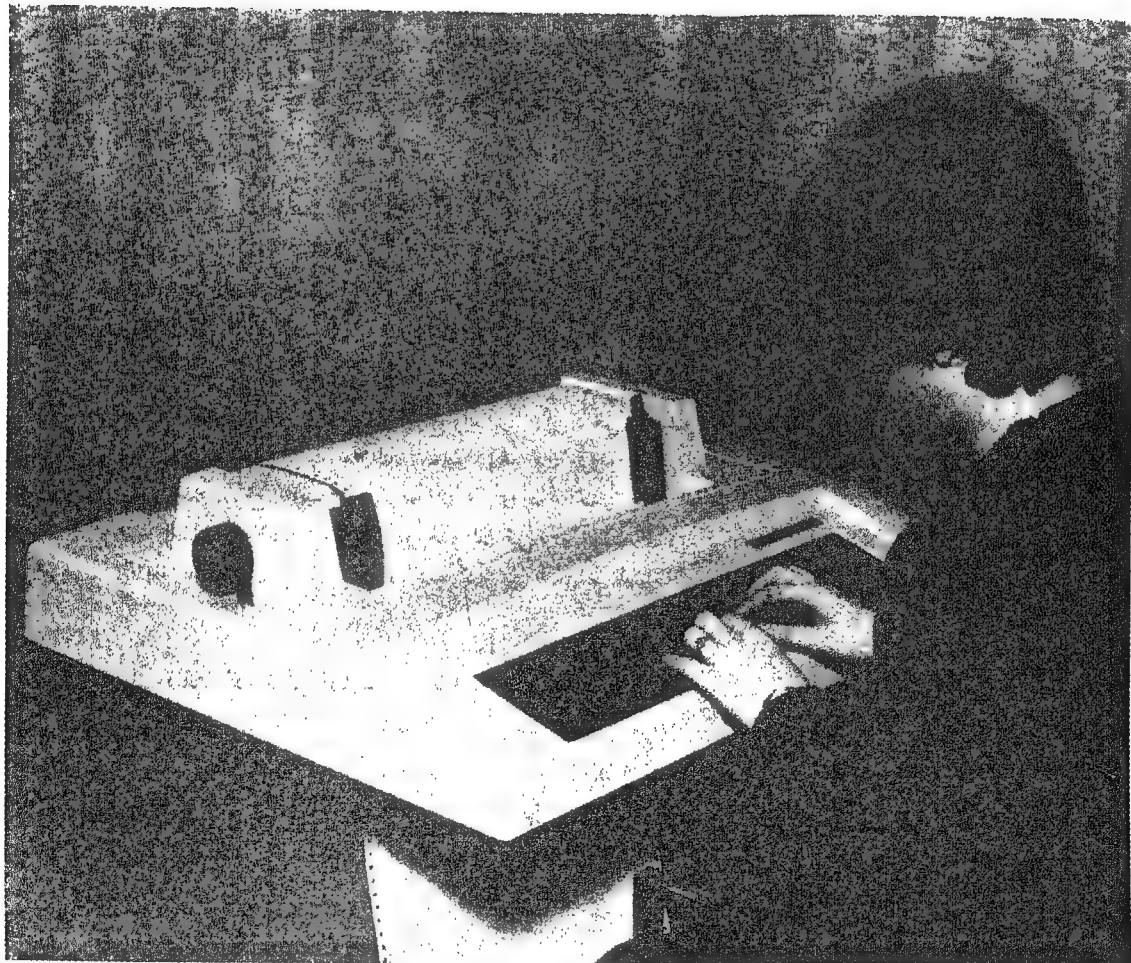
ويمكن نقل الكميات الكبيرة من المعلومات (البرامج والبيانات) إلى داخل الحاسب أو اخراجها بسرعة كبيرة في التشغيل على دفعات . هذا بالإضافة إلى أنه لا يلزم المستخدم أن يتواجد أثناء العملية ، ولهذا فإن طريقة العمل هذه تناسب العمليات التي تحتاج كميات كبيرة من وقت الحاسب أو الطويلة جداً . ومن ناحية أخرى رغم أنه في أى هذه العملية قد لا تحتاج إلى ثانية واحدة أو ثانيتين من الوقت الحقيقي للحاسب . (لا بد للعملية من الانتظار حتى يأتى دورها وذلك قبل قراءتها وتشغيلها وطباعتها) ولهذا فمن الممكن أن يكون التشغيل على دفعات غير مرغوب فيه إذا كانت هناك ضرورة لتشغيل عدد كبير من العمليات البسيطة والصغيرة والحصول على نتائجها بأسرع ما يمكن .

يمكن إرسال كميات كبيرة من المعلومات (برامج وبيانات) من وإلى الحاسب بسرعة فائقة في أسلوب التشغيل على دفعات . وهذا الأسلوب في التشغيل يناسب الشغلات الطويلة أو التي تتطلب وقتاً طويلاً من الحاسب . ومن الناحية الأخرى ، إجمالى الوقت المطلوب لتشغيل شغلة بهذه الطريقة يمكن أن يختلف من عدة دقائق إلى عدة ساعات رغم أن الشغلة تتطلب فعلاً ثانية أو اثنتين فقط من وقت الحاسب . (يجب أن تنتظر الشغلة دورها قبل أن تقرأ ثم تشغل وتطبع بعد ذلك) . وبذلك يكون التشغيل على دفعات غير مرغوب فيه عندما يكون المطلوب تشغيل عدة شغلات صغيرة وبسيطة وإرجاع النتائج بسرعة بقدر المستطاع .

الحساب التحوورى Interactive Computing

يمكن القيام بالحاسب التحوورى بحاسب شخصى صغير مثل المبين في شكل ١ - ١ أو بنهاية طرفية لحاسب كما هو موضح في شكل ١ - ٥ . وفي كلتا الحالتين يمد المستخدم الحاسب بالمدخلات من خلال لوحة مفاتيح وهي تشبه آلة كاتبة عادية . ثم تطبع النتائج المناظرة بعد ذلك إما على أفرخ كبيرة من الورق أو تعرض على وحدة TV . (يمكن أن تكون النتائج المطبوعة لازمة في عدة أنواع من التطبيقات ، حيث أنها تعطى « نسخة ملموسة » (أو وثيقة مطبوعة) للجلسة التحوورية . إلا أن استخدام العرض على شاشة عرض TV غالباً ما يكون أكثر ملاءمة) . أحياناً ما يشارك إلى النهايات الطرفية التحوورية للحاسبات . . .

الخاصية التي لها دلالة خاصة في التشغيل التحوورى للحاسب هي أن المستخدم والحاسب قادران على التحدث كل مع الآخر أثناء جلسة الحسابات . وبذلك يمكن أن يطلب من المستخدم بصفة دورية أن يمد الحاسب بمعلومات معينة تقرر نوع الإجراءات التالية التي يقوم الحاسب بتنفيذها .



شكل ١ - ٣

مثال ١ - ٤

يرغب طالب في استخدام حاسب شخصي لحساب نصف قطر دائرة مساحتها 100 . والمتاح هو برنامج يحسب مساحة الدائرة إذا أعطى نصف القطر . (لاحظ أن ذلك هو عكس الذي يرغبه الطالب تماماً) . وبالتالي فعلى الطالب أن يواصل بالمحاولة والخطأ ، تخمين قيمة لنصف القطر وتقرير ما إذا كانت النتيجة تناظر المساحة المطلوبة أم لا ، إن لم تكن فيفرض الطالب قيمة أخرى لنصف القطر ويحسب مساحة جديدة ، .. وهكذا .. وسوف يستمر في طريقة المحاولة والخطأ حتى يجد الطالب قيمة لنصف القطر تعطي مساحة قريبة جداً من 100 .

وحالما يدار الحاسب ويتم إدخال البرنامج فسوف تطبع الرسالة :

RADIUS = ?

وبعد ذلك يدخل الطالب قيمة لنصف القطر . دعنا نفرض أن الطالب أدخل قيمة 5 لنصف القطر . بعد ذلك يستجيب الحاسب بطباعة :

AREA = 87.5398

DO YOU WISH TO REPEAT THE CALCULATIONS ?

ثم يعطى الطالب نعم أو لا . إذا طبع الطالب نعم فسوف تعاد طباعة الرسالة :

RADIUS = ?

ويعاد الإجراء بالكامل . أما إذا طبع الطالب لا فتطبع الرسالة :

GOODBYE

وتنتهى الحسابات :

RADIUS=? 5
AREA= 78.5398

DO YOU WISH TO REPEAT THE CALCULATION? YES

RADIUS=? 6
AREA= 113.097

DO YOU WISH TO REPEAT THE CALCULATION? YES

RADIUS=? 5.6
AREA= 98.5204

DO YOU WISH TO REPEAT THE CALCULATION? NO

GOODBYE

نرى في شكل ١ - ٤ المعلومات التي تطبع خلال جلسة تفاعلية نموذجية ، مستخدماً البرنامج الموصوف أعلاه . وقد تم وضع خط تحت المعلومات المطبوعة بواسطة الطالب . تم التوصل إلى قيمة تقريبية فيها $r = 5.6$ بعد ثلاثة حسابات .

لاحظ كيف تبدو طريقة تحديث الحاسب والطالب كل مع الآخر . ولاحظ أيضاً أن الطالب ينتظر حتى يرى مساحة محسوبة قبل تقرير ما إذا كان سيقوم بعمل الحسابات مرة أخرى أم لا . وعلى

شكل ١ - ٤

ذلك ، فإن القيمة الجديدة التي يعطيها الطالب لنصف القطر محكومة بتأثير النتيجة السابقة التي تم حسابها .

يقال أحياناً أن البرامج التي يتم تصميمها للتطبيقات من النوع التفاعلي بأنها ذات طبيعة تفاعلية . الألعاب المبرمجة مثل أدفنشر — توتشكرز والشطرنج كلها أمثلة ممتازة لمثل هذه التطبيقات التفاعلية . وكذلك أيضاً الحركة السريعة للألعاب مثل غزاة الفضاء .

المشاركة الزمنية Timesharing

المشاركة الزمنية هي شكل من أشكال الحاسب التفاعلي حيث يستطيع عدد من المستخدمين استخدام حاسب واحد في نفس الوقت . يتصل كل مستفيد بالحاسب من خلال نهاية طرفية ، مثل المينة في شكل ١ - ٥ . يمكن أن تتصل النهايات الطرفية بالحاسب بواسطة سلك ، أو يمكن أن تتصل بالحاسب من خلال خطوط تليفونية أو دائرة موجات دقيقة . وبذلك فإن النهاية الطرفية للمشاركة الزمنية يمكن وضعها بعيداً — ربما على بعد عدة مئات من الأميال عن الحاسب المضيف لها .

حيث أن الحاسب يعمل أسرع بكثير من شخص جالس أمام نهاية طرفية فإن حاسباً واحداً يمكن أن يخدم عدداً كبيراً من النهايات الطرفية في نفس الوقت . وبذلك لا يبالي أى مستفيد بوجود مستفيدين آخرين ، ويبدو له أن الحاسب بالكامل يعمل من أجله فقط ونحت أمره . وعادة ما تستخدم الحاسبات الدقيقة مثل التي في شكل ١ - ١ بدلاً من النهاية الطرفية للمشاركة الزمنية . فقد أصبحت الوصلات شائعة على وجه الأخص من خلال خطوط تليفونية . ويسمح هذا التنظيم لشخص يعمل من منزله على حاسب شخصي أن يتصل بحاسب بعيد في المدرسة أو المكتب .

وتلائم المشاركة الزمنية تشغيل الشغلات البسيطة نسبياً والتي لا تتطلب نقل كميات كبيرة من البيانات أو تستهلك وقتاً طويلاً من الحاسب . وهي نفس الخصائص التي تحملها معظم تطبيقات الحاسب التي تبرز في المدارس والمكاتب التجارية . ويمكن تشغيل هذه التطبيقات سريعاً وببساطة وبأقل التكاليف باستخدام المشاركة الزمنية .

جامعة كبيرة بها حاسب وله إمكانيات المشاركة الزمنية ويتكون من 100 نهاية طرفية و 80 خط تليفون منفصل وتستخدم نظام المشاركة الزمنية وموضوعة في أماكن مختلفة من مبنى الجامعة . وتتصل هذه النهايات الطرفية بالحاسب الكبير عن طريق خطوط تليفونية . ترسل كل من هذه النهايات البيانات من وإلى الحاسب بسرعة قصوى مقدارها 120 حرف/ثانية . ويمكن استخدام كل النهايات الطرفية في آن واحد ، وبالرغم من ذلك فإنها تتفاعل مع حاسب واحد فقط . وتسمح الخطوط التليفونية للطلبة الغير موجودين بمبنى الجامعة بتوصيل حاسباتهم الشخصية بالحاسب الكبير ويمكن لكل حاسب شخصي أن ينقل بيانات من وإلى الحاسب الكبير بسرعة قصوى قدرها 120 حرف/ثانية وعلى ذلك فإنه يمكن للنهايات والحاسبات كلها وعددها 180 أن تتحاور مع الحاسب الرئيسى في وقت واحد . ومع كل فإن كل طالب لا يعلم أن الآخرين يستخدمون الحاسب في نفس الوقت .

وبالإضافة لذلك ، تم توصيل 20 نهاية طرفية عن بعد للحاسب . وضعت 15 نهاية طرفية منها في 5 مدارس ثانوية في المنطقة ، والنهايات الخمس الباقية وضعت في معمل أبحاث حكوى . جميع النهايات الطرفية وعددها 120 في آن واحد (وكثيراً ما يحدث ذلك) . وباقتسام الحاسب بهذه الطريقة يستطيع كل معهد أن ينتفع بخدمات الحاسب الكبير بتكلفة مناسبة .

يمكن - إذا تطلب الأمر ذلك - إدماج خصائص معينة من أسلوب التشغيل على دفعات وأسلوب المشاركة الزمنية ، فثلاً ، يمكن إدخال مجموعة بيانات مباشرة من النهاية الطرفية (وبذلك نكون في غنى عن التثقيب) ثم بعد ذلك يستكمل التشغيل بطريقة وأسلوب التشغيل على دفعات . ويمكن أيضاً استخدام قارئ البطاقات (تشغيل على دفعات) لإدخال برنامج ومجموعة بيانات ، ثم ينقح (يعدل) البرنامج وتشغل البيانات بأسلوب المشاركة الزمنية . مثل هذه العمليات الخليطة أصبحت أكثر شيوعاً كلما ازدادت أنظمة الحاسبات تعقيداً .

١ - ٤ أنواع لغات البرمجة TYPES OF PROGRAMMING LANGUAGES

يمكن استخدام عدة لغات مختلفة لبرمجة الحاسب . واللغة الأساسية هي لغة الآلة - وهي مجموعة تفصيلية من التعليمات المكونة والتي تتحكم في دوائر الحاسب الداخلية . وهذه هي اللهجة الطبيعية للحاسب . وقد تمت كتابة مجموعة قليلة من برامج الحاسب فعلاً بلغة الآلة ، ومع ذلك ، ولسبين هامين : أولهما ، أن لغة الآلة مرهقة جداً للعمل بها ، وثانيهما ، أن لكل حاسب مجموعة الأوامر الفريدة الخاصة به . (وبذلك فإن البرنامج المكتوب بلغة الآلة لنوع معين من الحاسبات لا يمكن تشغيله على نوع آخر من الحاسبات بدون تعديلات جوهرية) .

وعادة ، تم كتابة برامج الحاسب ببعض اللغات العالية المستوى حيث تتفق مجموعة الأوامر الخاصة بها مع لغات وأنكار الإنسان . معظم هذه اللغات العالية المستوى لغات لأغراض عامة مثل البيسك (بعض اللغات الأخرى الشائعة الاستخدام ولأغراض عامة أخرى هي Fortran, Cobol, Pascal و PL/I) . ويوجد أيضاً العديد من اللغات العالية المستوى لأغراض خاصة حيث تصمم مجموعة الأوامر الخاصة بها لأنواع معينة من التطبيقات .

وكقاعدة فإن الأمر الواحد في لغة عالية المستوى يكون مساوياً - لعدة أوامر من لغة الآلة . علاوة على ذلك ، فإن البرنامج المكتوب بلغة عالية المستوى يمكن تشغيله بصفة عامة على عدة أنواع مختلفة من الحاسبات بقليل من التعديلات أو بدون تعديلات على الإطلاق . ومعنى ذلك فإن اللغة العالية المستوى تقدم لنا بعض المزايا الهامة عن استخدام لغة-الآلة ألا وهي البساطة والتناسق والقابلية للنقل (أى الاستقلال عن الآلة) .

وبذلك فإن البرنامج المكتوب بلغة عالية المستوى يجب أن يترجم إلى لغة الآلة قبل أن ينفذ . وتعرف هذه المرحلة بالتفسير أو الترجمة ، معتداً في ذلك على الطريقة المعمول بها . (معظم نسخ البيسك يتم تفسيرها وليس ترجمتها) . البرامج المفسرة أسهل في العمل بها عن البرامج المترجمة بالرغم من أن البرامج المترجمة تنفذ أسرع بصفة عامة . ومع ذلك ففي كلتا الحالتين تنفذ

الترجمة أوتوماتيكياً بداخل الحاسب . وفي الحقيقة فإن المبرمج عديم الخبرة ربما لا يعي أن هذا الإجراء يتم ، حيث أنه يرى فقط البرنامج الأصل وبيانات الإدخال والنتائج وبيانات الإخراج .

المفسر والمترجم ما هو إلا برنامج للحاسب يقبل أى برنامج مكتوب بلغة عالية المستوى كبيانات إدخال ويولد برنامج مناظر مكتوب بلغة الآلة كخروج . وتبعاً لذلك ، فإن البرنامج الأصل المكتوب بلغة عالية المستوى يسمى برنامج المنبع ، والبرنامج الناتج المكتوب بلغة الآلة يسمى برنامج الهدف . ويجب أن يكون لكل حاسب مفسر أو مترجم لكل لغة عالية المستوى خاص بها . واستخدام المفسرات والمترجمات هو الذى يمكننا من الحصول على التناسق والاستقلال عن الآلة مع لغة عالية المستوى مثل لغة البيسك .

١ - ٥ مقدمة للغة البيسك INTRODUCTION TO BASIC

إن لغة البيسك سهلة في الاستخدام « مألوفة » تجمع أوامرها ما بين الصيغ الجبرية البسيطة وبعض الكلمات الدالة باللغة الإنجليزية مثل LET و READ و PRINT و GO TO و IF و THEN ،... إلخ . معظم اللغات العالية المستوى لها هياكل متشابهة ولكنها أصعب في التعلم والاستخدام عن لغة البيسك . وبذلك ، فإن البيسك تناسب بصفة عامة الأشخاص الذين يتعلمون البرمجة لأول مرة وفي الحقيقة فإن كثيراً من المدارس الثانوية والمدارس الإعدادية يعرض الآن دروساً في البرمجة بلغة البيسك ، كما أن عدداً من المدارس الأولية تقدم الآن الموضوع بصورة اختيارية لمجموعات من الطلبة .

إن استخدام البيسك مفيد بلون شك وغير مقصور في استخدامه للتطبيقات الأولية . يستخدم غالباً في عديد من التطبيقات المتقدمة المتنوعة في مجالات معينة مثل التجارة والاقتصاد وعلم النفس والطب وأيضاً في المجالات العلمية والهندسية والرياضيات . ولقد أصبح البيسك أيضاً اللغة الأولى « لهواة » الحاسبات الدقيقة الذين تتضمن اهتماماتهم ألعاب الحاسب التي تحتاج إلى استخدام الرسوم وتوليد الأصوات وكذلك التطبيقات التقليدية مثل الحسابات الشخصية وقواعد البيانات الإدارية .. إلخ وسوف نرى عينات مختارة لعدة أنواع مختلفة من التطبيقات في الأمثلة المبرمجة المتضمنة في هذا الكتاب .

نبذة تاريخية عن البيسك History of BASIC

تطور البيسك أساساً في كلية دارتموث الأمريكية بواسطة جون كيمنى وتوماس كورتر في منتصف الستينات . وقد تم اكتشافه سريعاً وتبناه عدد من تجار خدمات المشاركة الزمنية مما ساعد في انتشار اللغة وإلقاء الضوء عليها بين آلاف من مستخدمي الحاسبات . وبسرعة عرضت معظم الشركات المنتجة للحاسبات نسخ البيسك الخاصة بحاسباتهم . وبذلك أصبحت لغة البيسك بسرعة لغة المشاركة الزمنية ومن أكثر اللغات شيوعاً واستخداماً في الولايات المتحدة .

تلقى البيسك دفعة هامة مع تطور الحاسبات الدقيقة المنخفضة التكاليف التي بدأت في منتصف السبعينات . وحقيقة فإن كل الحاسبات الدقيقة قد تبنت البيسك كلغة برمجة أساسية لها ، كما أن عدداً من الحاسبات الدقيقة قد صممت مفسراً للغة البيسك كجزء دائم في دوائرها الداخلية . في الحقيقة ، فإن بساطة وملاءمة البيسك كان عاملاً أساسياً في التكاثر السريع لهذه الأجهزة . وأصبحت لغة البيسك متاحة في كل حاسب دقيق مباع ، وهناك العديد من الحاسبات الدقيقة تحتوي تصميم مفسر بييسك كجزء في دوائرها الداخلية .

في عام 1978 وضع المعهد القومى الأمريكى للمعايرة (ANSI) تقنينات لمجموعة فرعية مهمة من البيسك* ، من أجل زيادة التناسق بين نسخ البيسك وبعضها . ومع ذلك ، فإن معظم نسخ البيسك تحتوي على بعض الخصائص التي لا يتضمنها ANSI القياسى . وبذلك فيوجد بعض التنويعات في نسخ البيسك المختلفة المتاحة حالياً . وجار تطوير نسخة أكثر شمولاً من ANSI القياسى والتي سوف تتضمن هذه الخصائص الإضافية .

* American National Standard to Minimal BASIC, ANSI X 3.6 - 1978, American National Standards Institute, New York 1978.

التغيرات في البيسك Variations in BASIC

معظم نسخ البيسك المنفذة على الحاسبات الكبيرة أو المدعمة بواسطة خدمات المشاركة الزمنية يشبه بعضها الآخر . وكلها تتضمن الخواص الموضحة في النسخة المعيارية ANSI لعام ١٩٧٨ ، علاوة على عدد إضافي من الخواص الشائعة المستخدمة . وغالباً ما يشار إلى مثل هذه التفاوتات بتفاوتات البيسك الخاص بدارتموث . ويمكن لمعظم البرامج التي تنتفع بخواص البيسك الخاص بدارتموث أن تشغل على بعض الحاسبات المختلفة بتعديلات بسيطة أو بدون تعديلات على الإطلاق .

وقد تسببت المنافسة بين منتجي الحاسبات لإحباط أى محاولة في التقنيات الرسمية ، ويتم تعزيز معظم نسخ البيسك للحاسب الدقيق بأوامر خاصة تتفق مع امكانيات الأجهزة الخاصة بها . وعلى ذلك فإن هناك أوامر خاصة للقيام بإجراء الرسومات ، لتوليد الصوت والتحكم في الآلات المساعدة للحاسبات الدقيقة مثل الأقراص المرنة وآلات الطباعة والأقلام الضوئية وعصا التوجيه . وتشابه هذه الأوامر من نسخة بيسك حاسب دقيق إلى أخرى وعلى ذلك فإن العديد من برامج البيسك التي تكتب لأحد الحاسبات الدقيقة يمكن تعديلها لتعمل على حاسب دقيق آخر دون جهد كبير .

ولابد من اتباع هذه القواعد الست في أمثلة البرامج التي تعرض في أول ثمانية فصول في هذا الكتاب ، وعندما نصل إلى الفصل التاسع سنجد أنه يمكن التراجع عنها في بعض نسخ بيسك الحاسبات الدقيقة ، ومع كل فإن البرامج التي تطبق هذه القواعد ستظل صحيحة .

وقد تم في السنوات الأخيرة تطوير نسخ أكثر حداثة من البيسك تتضمن بعض الخصائص المعقدة المختلفة والتي لا توجد في النسخ التقليدية . ومن المدهش أن معظم هذه النسخ المعدلة من اللغة قد تم تطويرها للحاسبات الدقيقة وأن كان بعض هذه النسخ المعدلة من البيسك قد تم تطويرها أيضاً للحاسبات الكبيرة .

يصف هذا الكتاب كل الخصائص الشائعة في البيسك . ويتضمن العديد من الخصائص الأكثر حداثة والمعدلة بجانب الخصائص المعيارية الموجودة في العديد من النسخ التقليدية للغة . وبالتحديد ، فإن المواد الموجودة في الفصول من الثاني إلى السادس تطبق بانتظام على كل نسخ بيسك .

تتضمن أيضاً معظم نسخ البيسك للحاسبات الكبيرة الخواص الموضحة في الفصل السابع (المتجهات والمصفوفات) . وتطبق المواد الموجودة في الفصل الثامن (ملفات البيانات) على أعداد كبيرة من نسخ البيسك ، بالرغم من وجود اختلافات يجب أخذها في الاعتبار بالطريقة الصحيحة عند تنفيذ هذه الخواص . وأخيراً تصف الفصول من التاسع إلى الثاني عشر العديد من الخصائص الخاصة والطرق الفنية الموجودة في بيسك الحاسبات الدقيقة مع التأكيد الخاص على ميكروسوفت بيسك المنفذ على حاسب IMB الشخصي ويحتوى الكتاب أيضاً على خمس ملاحق والتي تلخص معظم المادة التي سبق عرضها .

ولن يواجه القارئ المتكمن من المادة العلمية أى صعوبة في استغلال الخواص المتاحة في نسخة معينة من البيسك ، أو في تغيير برنامج حتى يمكن تشغيله مع نسخة مختلفة من اللغة .

بناء برنامج البيسك Structure of a BASIC Program

يكتب كل أمر في برنامج البيسك كجملة منفصلة . وبذلك فإن برنامج البيسك الكامل سوف يتكون من جمل متتالية . يجب أن تظهر هذه الجمل بالترتيب الذي يجب أن تنفذ به إلا إذا نمت الإشارة « للقفز » عن قصد (أى تحويل التحكم) .

تطبق القواعد التالية على جمل البيسك :

١ - يجب ظهور كل جملة على سطر منفصل (*) .

٢ - لا يمكن أن تتعدى الجملة طول سطر واحد (أى لا يمكن « استكمالها » من سطر لآخر) .

* على معظم النهايات الطرفية للحاسب ، السطر يساوى 80 عموداً ؛ ومع ذلك ، فبعض النهايات الطرفية تسمح بعدد من الحروف يصل إلى 132 حرفاً / سطر .

٣ - يجب أن تبدأ كل جملة بكلمة صحيحة موجبة تعرف كرقم الجملة (أو رقم السطر) .

٤ - يجب أن تكون لجملة المتعاقبة أرقام سطور متزايدة .

٥ - يجب أن يتبع كل رقم سطر كلمة دالة من كلمات البيسك تشير لنوع الأمر الذى يجب القيام به .

٦ - يمكن إضافة الفراغات فى أى مكان مطلوب من أجل تحسين قراءة الجملة .

يمكن أيضاً تفصيل السطور الفارغة (الخالية) فى برنامج البيسك . مع ذلك ، فإن كل سطر خال يجب أن يحمل رقم سطر فريد ، ويتبع رقم السطر مسافة خالية واحدة على الأقل . (أما إذا تلى رقم السطر رجوع عربة الآلة فوراً . فإن رقم السطر هذا سوف يلغى من البرنامج . وسوف يتم مناقشة ذلك فى الفصل الثالث) .

ولابد من اتباع هذه القواعد الست فى أمثلة البرامج التى تعرض فى أول ثمانية فصول من هذا الكتاب ، وعندما نصل إلى الفصل التاسع سنجد أنه يمكن التراخى عنها فى بعض نسخ بيسك الحاسب الدقيق ، ومع كل فإن البرامج التى تطبق هذه القواعد ستظل صحيحة .

مثال ١ - ٦ مساحة دائرة

يمثل شكل ١ - ٥ برنامج بيسك بسيط لحساب مساحة دائرة معروف نصف قطرها . والمنطق المستخدم لإجراء هذه الحسابات قد تم مناقشته فى مثال ١ - ١ . إن البرنامج بسيط للغاية ، ومع ذلك فإن الأساس المنطقى يمكن تحديده بواسطة عملية فحص بسيط .

```
10 INPUT R
20 LET A=3.14159*R^2
30 PRINT R,A
40 END
```

نرى أن البرنامج يتكون من أربع جمل كل منها يظهر فى سطر منفصل . كل جملة لها رقم جملة خاص بها (أو رقم سطر) . وتزيد هذه الأرقام بالتعاقب من بداية (أعلى) إلى نهاية (أسفل) البرنامج . تحتوى الجملة على كلمات بيسك الدالة وهى END و PRINT و LET و INPUT بالترتيب

شكل ١ - ٥

الغرض من الجملة الأولى (INPUT R) هو إدخال قيمة رقمية لنصف القطر R من لوحة التشغيل المركزية . وتسبب الجملة الثانية (LET A=3.14159*R^2) فى إيجاد قيمة R² . ثم بعد ذلك سوف تمثل هذه الكمية بالحرف A (للمساحة) . وتسبب الجملة الثانية (PRINT R,A) فى إرسال القيم الرقمية R و A للوحة التشغيل المركزية ، حيث يتم طباعتها بعد ذلك . وأخيراً ، فإن آخر جملة (END) مطلوبة للتميز عن انتهاء البرنامج .

لاحظ الرموز المستخدمة فى السطر 20 للتمييز عن العمليات الحسابية . فيشار إلى الضرب بواسطة نجمة (*) ، ويستخدم السهم الرأسى (↑) لرفع كمية لقوة معينة . (هذه العملية الأخيرة معروفة كعملية الأس وتستخدم بعض الطرفيات علامة ^) للتمييز عن الأس كما هو مبين بشكل ١ - ٥ . وتمثل العمليات الحسابية الأخرى والمنسجمة بالجمع والطرح والقسمة فى البيسك بواسطة علامة الجمع (+) وعلامة الطرح (-) والخط المائل (/) على الترتيب .

Some Advantages of BASIC بعض مزايا لغة البيسك

١ - إن لغة البيسك لغة « صديقة » أى « مناسبة للإنسان » . فهى سهلة للتعلم ومسلية فى الاستخدام . أى شخص منظم يمكنه تعلم كيف يبرمج بلغة البيسك وليس من الضروري أن يكون عنده خلفية كبيرة فى الرياضيات .

٢ - إن اللغة غاية فى المرونة ، وتسمح للمبرمج بتطوير برامج جديدة والتمييز فى البرامج الموجودة . بجهود قليلة نسبياً .

- ٣ - تناسب لغة البيسك استخدام البيئة التحوارية . وهذا يتضمن التطبيقات المخصصة للحاسبات الدقيقة بجانب تطبيقات المشاركة الزمنية للحاسبات الكبيرة .
- ٤ - إن اللغة متاحة عالمياً على كل من الحاسبات الكبيرة والصغيرة . وقد أصبحت لغة البرمجة القياسية لمعظم تطبيقات الحاسبات الدقيقة .
- ٥ - إن الخواص الشائعة الاستخدام في البيسك قياسية نسبياً ، بالرغم من احتمال وجود فروق بسيطة بين نسخة من البيسك وأخرى . وبذلك فإن اللغة مستقلة عن الآلة استقلالاً كبيراً . وبالتالي ، فإن معظم برامج البيسك يمكن تشغيلها على حاسبات مختلفة كثيرة بقليل من التعديلات أو بدون تعديلات إطلاقاً .

اسئلة للمراجعة

Review Questions

- ١-١ ما هو المقصود بحاسب كبير ؟ وأين توجد الحاسبات الكبيرة وفيما تستخدم عادة ؟
- ٢-١ ما هو الحاسب الصغير ؟ وكيف يختلف الحاسب الصغير عن الحاسب الكبير ؟
- ٣-١ ما هو الحاسب الدقيق ؟ وكيف تختلف الحاسبات الدقيقة عن الحاسبات الكبيرة والحاسبات الصغيرة .
- ٤-١ اذكر نوعين مختلفين من البيانات .
- ٥-١ ما هو المقصود ببرنامج الحاسب ؟ وماذا يحدث ، وعموماً ، عند تنفيذ برنامج الحاسب ؟
- ٦-١ ما هي ذاكرة الحاسب ؟ ما هي أنواع المعلومات التي تخزن في ذاكرة الحاسب ؟
- ٧-١ ما هو الرقم الثنائي ؟ ما هو البايت ؟ ما هو الفرق بين البايت والكلمة في الذاكرة ؟
- ٨-١ ما هو المصطلح المستخدم لوصف حجم ذاكرة الحاسب ؟ ما هي بعض الأحجام النموذجية للذاكرة ؟
- ٩-١ اذكر أسماء بعض الأجهزة النموذجية للذاكرة المساعدة . وكيف يختلف هذا النوع من الذاكرة عن الذاكرة الأساسية للحاسب ؟
- ١٠-١ ما هي وحدة قياس الزمن المستخدمة للتعبير عن سرعة المهام الأساسية التي يؤديها الحاسب ؟
- ١١-١ ما هو الفرق بين أسلوب التشغيل على دفعات وأسلوب التشغيل التحواري ؟ ما هي مميزاتها وعيوبها ؟
- ١٢-١ ما هو المقصود بالمشاركة الزمنية ؟ ما هي أنواع التطبيقات التي تناسبها المشاركة الزمنية ؟
- ١٣-١ ما هي لغة الآلة ؟ وكيف تختلف عن لغات المستوى العالي ؟
- ١٤-١ اذكر بعض لغات مستوى عال شائعة الاستخدام . وما هي مميزات استخدام لغات المستوى العالي ؟
- ١٥-١ ما هو المقصود بالتفسير أو الترجمة وكيف تختلف العمليتان ؟
- ١٦-١ ما هو برنامج المنبع ، برنامج الهدف ، ولماذا تكون هذه المفاهيم مهمة ؟

- ١٧-١ ماذا تعنى كلمة البيسك ؟
- ١٨-١ ما هى الخواص العامة للغة البيسك ؟
- ١٩-١ أين تم وضع لغة البيسك ؟ ومن هو الذى قام بذلك ؟
- ٢٠-١ إلى أى مدى تختلف كل نسخة مستقلة من البيسك عن أخرى ؟ وهل تم توحيد معيارى اللغة ؟
- ٢١-١ ما هى جملة البيسك وبأى ترتيب يجب ظهور الجمل فى برنامج البيسك ؟
- ٢٢-١ لخص القواعد الست التى تطبق على كل جمل البيسك . وهل هذه القواعد تطبق فى معظم نسخ بيسك الحاسب الدقيق .
- ٢٣-١ ما هى الرموز التى تستخدم فى البيسك لتشير إلى الجمع والطرح والضرب والقسمة ؟
- ٢٤-١ ما هو المقصود بالأس ؟ وما هو الرمز المستخدم فى البيسك لتمثيل الأس ؟
- ٢٥-١ لخص المزايا الأساسية للغة البيسك .

مسائل محلولة

Solved Problems

- ٢٦-١ عدة برامج بيسك أولية معروضة فيما يلى . اشرح غرض كل برنامج منها

```
10 INPUT L,W
20 LET A=L*W
30 PRINT L,W,A
40 END
```

(أ)

يحسب هذا البرنامج مساحة مستطيل معروف طول وعرضه .

(ب)

```
10 INPUT A,B,C,D,E
20 LET S=A+B+C+D+E
30 PRINT A,B,C,D,E
40 PRINT S
50 END
```

يحسب هذا البرنامج حاصل جمع 5 أعداد . لاحظ أن الأعداد الخمسة سوف تطبع على سطر واحد ويطبع حاصل الجمع المحسوب على السطر التالى . (كل جملة من جمل الطباعة PRINT تبدأ من سطر جديد) .

(ج)

```

10 INPUT A,B,C
20 LET X1=(-B+(B↑2-4*A*C)↑.5)/(2*A)
30 LET X2=(-B-(B↑2-4*A*C)↑.5)/(2*A)
40 PRINT A,B,C,X1,X2
50 END

```

يحسب هذا البرنامج قيم x_1 و x_2 من الصيغ الرياضية

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

حيث تحدد قيم a و b و c

٢٧-١ اكتب برنامج ييسك أول لكل من الحالات الموصوفة فيما يلي :

(١) احسب نصف قطر دائرة معروفة المساحة (انظر مثال ١-٧) .

بما أن $A = \pi r^2$ فيمكننا حلها لإيجاد قيمة r ، بحيث تكون

$$r = \sqrt{A/\pi}$$

وبذلك فإن البرنامج المطلوب هو :

```

10 INPUT A
20 LET R=(A/3.141593)↑.5
30 PRINT A,R
40 END

```

(ب) احسب طول مستطيل معروف المساحة والعرض (انظر المسألة ١-٢٦) .

```

10 INPUT A,W
20 LET L=A/W
30 PRINT A,W,L
40 END

```

(ج) احسب حاصل ضرب 5 أرقام معطاة (انظر مسألة ١-٢٦ ب) .

```

10 INPUT A,B,C,D,E
20 LET P=A*B*C*D*E
30 PRINT A,B,C,D,E
40 PRINT P
50 END

```


مسائل تكميلية (اضافية)
Supplementary Problems

٢٨ - ١ نعرض فيما يلي عدة برامج أساسية للبيسك . اشرح الغرض من كل برنامج .

```
10 INPUT B,H
20 LET A=(B*H)/2
30 PRINT B,H,A
40 END
```

(ا)

```
10 INPUT L,W,
20 LET C=2*(L+W)
30 PRINT L,W,C
40 END
```

(ب)

```
10 INPUT U,V
20 LET W=U+V
30 LET X=U-V
40 LET Y=U*V
50 LET Z=U/V
60 PRINT U,V
70 PRINT W,X,Y,Z
80 END
```

(ج)

```
10 INPUT X
20 LET Y=1+X+(X^2)/2+(X^3)/6
30 PRINT X,Y
40 END
```

(د)

٢٩ - ١ اكتب برنامج بيسك أول لكل من الحالات الموصوفة فيما يلي :

(ا) احسب محيط دائرة إذا أعطى نصف قطرها (انظر مثال ١ - ٧) .

(ب) احسب طول وتر المثلث القائم الزاوية إذا أعطيت كلا من القاعدة والارتفاع (انظر المسألة (١ - ٢٨)) .

(ج) احسب قيمة الصيغة الرياضية

$$w = \frac{u-v}{u+v}$$

حيث تعطى قيم u ، v (انظر المسألة (١ - ٢٨)) .

(د) احسب قيمة الصيغة الرياضية :

$$y = 100(1 + x + 2x^2 + 3x^3)$$

حيث توصف قيمة x (انظر المسألة (١-٢٨ د)).

١-٣٠ فيمايل برنامج بيك لحساب مساحة ومحيط مستطيل وكذا طول القطر . ولكن بعض الجمل مكتوبة بطريقة غير صحيحة . حدد جميع الأخطاء .

```
10 INPUT L,W
20 LET A=L*W      30 LET P=2*(L+W)
35 D=(L↑2+W↑2)↑.5
25 PRINT L,W,A,P,D
40 END
```

الفصل ٢

طريق البداية للغة البيسك

Getting Started with BASIC

سوف نفحص في هذا الفصل عدة مفاهيم جوهرية للغة البيسك ، مثل الأرقام ، والمتغيرات والصيغ الرياضية . ثم بعد ذلك سوف ندرس الست جمل الأكثر شيوعاً عند استخدام البيسك ، والتي تسمح لنا بعمل إدخال / إخراج وعملية تداول البيانات والقفز إلى أجزاء أخرى من البرنامج عندما نرغب ذلك . وبعد الانتهاء من هذا الفصل سوف يكون القارئ قادراً على كتابة برامج بيسك خاصة به لمسائل مختلفة .

٢ - ١ الأرقام (الثوابت) (NUMBERS (CONSTANTS)

يشار للكميات الرقمية في البيسك كأرقام (أو ثوابت) ويمكن التعبير عن الأرقام بطريقتين مختلفتين : كميات صحيحة (أرقام صحيحة بدون علامة عشرية) أو كميات عشرية (أرقام بها علامة عشرية) . تطبق القواعد الآتية على كتابة الأرقام :

- ١ - لا يمكن للفصلة أن تظهر في أى مكان من الرقم .
- ٢ - يمكن أن يسبق الرقم علامة + أو - (ويفهم أن الرقم موجب إذا لم تظهر أى علامة) .
- ٣ - يمكن أن يحتوى الرقم على أس ، إذا نطلب ذلك . يشبه التمثيل الأسى التمثيل العلمى ، ماعداً أن الأساس 10 يستبدل بالحرف E وبذلك فإن الكمية 1.2×10^{-3} يمكن كتابتها في البيسك $1.2 \text{ E}-3$ ويمكن للأس أن يكون إما موجباً أو سالباً ولكن لا يمكن أن يحتوى على علامة عشرية .
- ٤ - معظم نسخ البيسك تسمح أن يصل الرقم إلى 8 أو 9 خانات معنوية .
- ٥ - يمكن أن تكون القيمة كبيرة لتصل إلى 10^{38} أو صغيرة لتصل إلى 10^{-38} (وتختلف هذه الكمية من نسخة بيسك لأخرى) ويسمح أيضاً بالصفر .

مثال ٢ - ١

تعبّر الكميات المدددة التالية عن أرقام مقبولة في البيسك . لاحظ أن كل كمية (كل صف) يمكن أن تكتب بعدة طرق مختلفة .

0	+0	-0
1	+1	0.1E+1
-5280	-5.280E+3	-.5280E4
+1492	1492	1.492E+3
-.0000613	-6.13 E-5	-613 E-7
3000000	3E6	3 E+6

٢ - ٢ سلاسل الحروف (STRINGS

سلسلة الحروف هي عدة حروف متتالية (أى حروف ، أعداد ، حروف خاصة معينة مثل + و - و / و * و = و \$ و ... الخ) . يمكن أن تتضمن سلسلة الحروف أماكن خالية ولكن لا يمكن أن تتضمن علامات الاقتباس . وأقصى عدد يمكن أن تتضمنه سلسلة الحروف يختلف من نسخة بيسك لأخرى . في بعض النسخ لا يمكن أن تمتد سلسلة الحروف 15 حرفاً ، بينما تسمح نسخ أخرى بعدد يصل إلى 4095 حرفاً .

تستخدم سلاسل الحروف لتمثيل المعلومات غير الرقية مثل أسماء وعناوين و . . . الخ . وهي تستخدم أيضاً لعنونة بيانات المخرجات العددية ولطباعة الرسائل النصية (التطبيقات التي سوف نراها في الكتاب بعد ذلك) .

مثال ٢ - ٢

فيما يلي عدة سلاسل الحروف

SANTA CLAUS	TYPE A VALUE FOR C:
APOLLO-17	\$19.95
X1=	3730425
DO YOU WISH TO TRY AGAIN?	THE ANSWER IS

لاحظ أن الأرقام الصحيحة المتتالية مثل 3730425 لا تمثل كمية عددية إذا كتبت كسلسلة من الحروف .

٢ - ٣ المتغيرات VARIABLES

المتغير هو اسم يمثل رقماً أو سلسلة حروف . وفي نسخ بيسك القديمة كل متغير عددي يجب أن يتكون من حرف أو حرف يتبعه رقم صحيح . المتغير الحرفي يجب أن يكتب كحرف يتبعه علامة الدولار (\$) . وتسمح أيضاً معظم نسخ البيسك للمتغير الحرفي أن يكتب في صورة حرف يتبعه رقم صحيح ويتبعه علامة الدولار \$.

مثال ٢ - ٣

يمكن أن تمثل كل من المتغيرات التالية كميات عددية :

A K X C1 X5

يمكن أن تمثل كل من المتغيرات التالية سلسلة من الحروف :

A\$ K\$ X\$ C\$ T\$

في معظم نسخ البيسك يمكن أن تمثل سلسلة الحروف بواسطة أي من المتغيرات التالية :

A2\$ K9\$ X0\$ C1\$ X5\$

٢ - ٤ المعاملات والصيغ الرياضية (التعبيرات الرياضية)

OPERATORS AND FORMULAS (EXPRESSIONS)

تستخدم في البيسك رموزاً خاصة تسمى معاملات وذلك لتشير للعمليات الحسابية الجمع والطرح والضرب والقسمة والأس . هذه المعاملات هي :

(علامة الجمع)	+	الجمع
(علامة الطرح)	-	الطرح
(نجمة)	*	الضرب
(شرطة مائلة)	/	القسمة
(سهم رأسى يشير إلى أعلى)	↑	الأس

(بعض النهايات الطرفية تستخدم الرمز المميز)

تستخدم المعاملات لتصل بين الأرقام والمتغيرات العددية ، وبذلك تكون صيغاً رياضية (أو تعبيرات رياضية) .

وتجرى العمليات المشار إليها على الحدود الرقبة في الصيغ الرياضية وتنتج قيمة عددية وحيدة . ومن ثم فإن الصيغ الرياضية تمثل كمية رقبة محدودة .

مثال ٢ - ٤

نعرض فيما يلي عدة صيغ رياضية للغة ببلك :

$$\begin{aligned} J+1 \\ A+B-C \\ (2*X-8*Y)/(U+V) \\ 3.141593*R+2 \\ B+2-4*A*C \end{aligned}$$

تمثل كل صيغة كمية رقبة . وبذلك إذا كانت المتغيرات A و B و C تمثل الكميات الرقبة 2 و 5 و 3 على الترتيب فسوف تمثل الصيغة الرياضية $A + B - C$ بالكمية 4 .

وبالتحديد ، فإن الصيغة الرياضية يمكن أن تتكون من عدد وحيد أو متغير رقمي وحيد ، أو توافقية من الأعداد والمتغيرات العددية والمعاملات . ومع ذلك ، فن المهم أن تفهم أن المتغير العددي يجب أن تحدد قيمته بكمية رقبة قبل أن يظهر في الصيغة الرياضية . وإلا فلا يمكن حساب قيمة الصيغة الرياضية لتنتج قيمة عددية .

٢ - ٥ التدرج الهرمي للعمليات الحسابية HIERARCHY OF OPERATIONS

إذا كان هناك معاملا أو أكثر في أي صيغة رياضية ، يمكن أن تبرز أسئلة بمعنى هل تناظر الصيغة الرياضية $2 * X - 3 * Y$ للحد الجبري $(2x) - (3y)$ أو $2(x - 3y)$ ؟ وبالمثل هل تناظر $A/B * C$ لكمة $a/(bc)$ أو $(a/b)C$ ؟ يمكن أن يجاب عن هذه الأسئلة بسهولة إذا عرفنا التدرج الهرمي للعمليات الحسابية ومرتبة التنفيذ بداخل مجموعة متدرجة .

التدرج الهرمي للعمليات الحسابية هو :

١ - الأس تؤدي كل عمليات الأس أولا .

٢ - الضرب والقسمة تجرى هذه العمليات الحسابية بعد أن تنتهي من تأدية كل العمليات الأسية . وليس من الضروري أن يسبق الضرب القسمة .

٣ - الجمع والطرح هذه آخر العمليات التي تقوم بها . وليس من الضروري أن يسبق الجمع الطرح .

تجرى هذه العمليات الحسابية بداخل مجموعة متدرجة من اليسار إلى اليمين .

مثال ٢ - ٥

الصيغة الرياضية :

$$A/B * C$$

مساوية للتعبير الرياضي $(a / b)c$ ، حيث أننا نقوم بالعمليات الحسابية من اليسار إلى اليمين .

وبالمثل الصيغة الرياضية :

$$B+2-4*A*C$$

مساوية للتعبير الرياضي $b^2 - (4ac)$ في هذه الحالة تجرى $2 \uparrow B$ أولا ثم يتبعها حاصل الضرب $(4 * A)$ أولا ثم بعد ذلك $(4*A)*C$ ثم تجرى عملية الطرح أخيراً منتجة الكمية :

$$(B+2)-(4*A*C).$$

٢ - ٦ استخدام الأقواس USE OF PARENTHESES

هناك عدة مواقف تتطلب تغيير التدرج الطبيعي للمعاملات الحسابية في أى صيغة رياضية . ويمكن أن يتم إنجاز ذلك بسهولة بإضافة زوج من الأقواس في الأماكن الصحيحة بداخل الصيغة الرياضية . وتنفذ المعاملات الحسابية بداخل الأقواس الداخلية أولاً تتبعها المعاملات الحسابية بداخل الأقواس الخارجية ، . . . وهكذا . ولكن سوف يطبق التدرج الطبيعي بداخل زوج من الأقواس إلا إذا تم تعديله بواسطة أزواج أخرى من الأقواس مبيتة داخل الزوج المعطى .

يجب أن نتذكر دائماً استخدام أزواج من الأقواس . إهمال التوازن في الأقواس الجبى أو اليسرى من الأخطاء الشائعة .

مثال ٢ - ٦

نفرض أننا نرغب في حساب قيمة :

$$[2(a + b)^2 + (3c)^2]^{m/(n+1)}$$

فإن صيغة البيسك الرياضية التى تناظر هذا الحد الجبرى هى :

$$(2*(A+B)^2+(3*C)^2)^(M/(N+1))$$

إذا كان هناك شك في ترتيب تنفيذ المعاملات الحسابية فيمكننا تقديم زوج إضافي من الأقواس ، وذلك يعطى :

$$((2*((A+B)^2))+((3*C)^2))^(M/(N+1))$$

كلتا الصيغتين الرياضيتين صحيح . ولذلك فيمكن أن نفضل الصيغة الرياضية الأولى ، حيث أنها أقل عدداً في الأقواس وبالتالي أسهل في القراءة

٢ - ٧ قواعد خاصة متعلقة بالصيغ الرياضية SPECIAL RULES CONCERNING FORMULAS

يمكن أن تبرز مشاكل خاصة إذا لم تكتب الصيغة الرياضية صحيحة . ويمكن تجنب ذلك إذا طبقنا هذه القواعد :

١ - استهلال المتغير بعلامة (-) مساو لضربه في 1 -

مثال ٢ - ٧

الصيغة الرياضية :

$$-X+N$$

مساوية للكية $-(X+N)$ أو $-1*(X+N)$ ، وحيث أن الأس له أسبقية على الضرب . ومن ثم إذا كانت قيمة X و N هى 2 و 3 على الترتيب . فإن $-X+N$ سوف تنتج القيمة 9 - .

٢ - ماعداً الشروط التى تم ذكرها ، فإن المعاملات الحسابية لا يمكن أن توصف بطريقة ضمنية .

مثال ٢ - ٨

يجب أن يكتب التعبير الجبرى $(x_1 + 3x_2)$ في البيسك (BASIC) كالتالى :

$$2*(X1+3*X2)$$

بتوضيح تام لمعامل الضرب . الصيغتان الرياضيتان $2*(X1 + 3 * X2)$ و $2 (X1 + 3 * X2)$ غير صحيحتين .

٣ - يمكن رفع كية سالبة لقوة إذا كان الأس رقماً صحيحاً (يجب ألا يختلط عليك الأمر بين الأس كصيغة رياضية أسية والأس كجزء من الرقم العشرى) .

ولفهم هذا التقييد ، يجب أن نرى كيف تتم عملية الأس . إذا كان الأس كمية صحيحة فإن الكمية المرفوعة للأس تضرب في نفسها عدداً مناسباً من المرات .

ومن ناحية أخرى ، نفرض أن الأس كمية عشرية . فإن الإجراء المتبع مع الأس العشري هو حساب اللوغاريتم للكمية المطلوبة أن ترفع للأس ثم ضرب هذا اللوغاريتم في الأس ثم حساب مقابل اللوغاريتم . وحيث أن لوغاريتم الرقم السالب غير معرف ، فإننا نرى أن العملية غير مسموح بها إذا كانت الكمية المطلوب رفعها لأس هي كمية سالبة .

مثال ٢ - ٩

أدرس الصيغة الرياضية :

$$(C1+C2)^{\uparrow 3}$$

تضرب الكمية المثلثة $(C1 + C2)$ في نفسها ثلاث مرات وبذلك ينتج مكعب الكمية . لاحظ أنه لا يهم إذا كانت الكمية $(C1 + C2)$ سالبة أو موجبة

ومع ذلك ، فإن الصيغة الرياضية :

$$(B^{\uparrow 2}-4*A*C)^{\uparrow .5}$$

سوف تكون صحيحة ومقبولة في حالة ما إذا كانت $B^{\uparrow 2}-4*A*C$ تمثل بكمية موجبة فقط .

وأخيراً ، إدرس ماذا يحدث إذا كان أي من A أو N في التعبير $A^{\uparrow N}$ مساوياً لصفر . إذا كانت N تساوى صفرًا فإن $A^{\uparrow N}$ سوف تكون قيمتها 1 دون الاكثارات بقيمة A . وإذا كانت قيمة A تساوى صفرًا و N قيمة غير الصفر فإن قيمة $A^{\uparrow N}$ تساوى صفرًا .

٤ - لا يمكن إجراء أى عمليات حسابية على سلاسل الحروف أو المتغيرات الحرفية ، ومع ذلك فإن بعض نسخ البيسك تسمح لسلسلة وسلسلة متغيرات أن تتصلا تعاقبياً (أى تتجمع واحدة خلف الأخرى) .

مثال ٢ - ١٠ :

افترض أن المتغيرين غير الرقميين $X\$$, $Y\$$ قد خصص لهما القيم الآتية :

$X\$ = "TEN"$

$Y\$ = "THOUSAND"$

$X\$ + Y\$ + "DOLLARS"$

لا تتحقق حيث أنها ليست ذات معنى لتجرى عمليات عددية على سلاسل الحروف . ويكون التعبير في بعض نسخ البيسك .

$X\$ + " " + Y\$ + "DOLLARS"$

سوف يتسبب في ضم السلاسل الحرفية الثلاث ، وتنتج سلسلة حروف مفردة .

TEN THOUSAND DOLLARS

٨ - ٢ تحديد قيم — جملة LET ASSIGNING VALUES—THE LET STATEMENT

تستخدم جملة *LET* لتحديد قيمة رقمية أو قيمة حرفية لمتغير . يمكننا تعريف متغير معين في البرنامج بواسطة تحديد قيمته بهذه الطريقة .

تتكون جملة LET من رقم جملة متبوعة بالكلمة الدالة LET متبوعة بمصطلح تحديد يشبه المعادلة الرياضية . يجب أن يتكون مصطلح التحديد من متغير وعلامة (=) وصيغة رياضية كما هو موضح في الأمثلة التالية .

مثال ٢ - ١١

```
10 LET X=12.5
20 LET C1=F3
30 LET A=3.141593*R↑2
40 LET N$="NAME"
50 LET T$=N$
```

في كل من هذه الجمل تعطى قيمة الحد الموجود يمين علامة التساوى للمتغير الموجود على يسار علامة التساوى .

لاحظ أن المتغير الموجود على يسار علامة التساوى والحد الموجود على يمين العلامة يجب أن يكونا من نفس النوع (أما عددية أو حرفية) . وبمعنى آخر ، فإن القيمة العددية لا يمكن أن نحدد بها قيمة متغير حرفي والعكس صحيح . لاحظ أيضاً أن سلسلة الحروف يجب أن توضع بين علامتي اقتباس إذا ظهرت في جملة LET .

من الأهمية أن نفهم الفرق بين مصطلح التحديد الذي يظهر في جملة LET والمعادلة الجبرية . تبدو العديد من مصطلحات التحديد كمعادلات جبرية ومن الناحية الأخرى ، فإن هناك بعض مصطلحات التحديد الجائزة والتي ليس لها أى معنى إذا عوملت معاملة المعادلات الجبرية .

مثال ٢ - ١٢

إدرس جملة LET التالية ، وهي صحيحة وذات معنى :

```
5 LET J = J + 1
```

واضح أن مصطلح التحديد $J = J + 1$ لا يناظر معادلة جبرية حيث المعادلة $J = J + 1$ ليس لها معنى . ومانعمله هنا هو زيادة قيمة المتغير الرقى J بالوحدة . وبذلك فإن مصطلح التحديد منطقي تماماً إذا فسرناه كالتالى : أضف واحداً للقيمة التي يمثلها المتغير J ثم ضع القيمة الجديدة في J . لاحظ أن القيمة القديمة للمتغير J سوف تستبدل بالقيمة الجديدة .

سوف ترى أن جمل LET من هذا النوع تستخدم غالباً في البيسك .

تسمح بعض نسخ البيسك بمرونة أكثر في كتابة جمل LET عن نسخ أخرى . فمثلاً ، في بعض نسخ البيسك (BASIC) يمكن تحديد نفس القيمة لمتغيرين أو أكثر في جملة LET واحدة . علاوة على ذلك ، فيمكن حذف الكلمة الدالة LET في نسخ معينة من البيسك .

مثال ٢ - ١٣

يمكن لجمل LET التالية أن يكون مسموحاً بها في بعض نسخ البيسك :

```
10 LET A=B=C=5.089
20 A=L*W
30 X1=X2=(A+B)/(C+D)
40 LET A$=K$="TERMINATE"
```

لاحظ أن الجملة الأولى والثالثة والرابعة تتضمن تحديد قيم مضاعفة وأن الكلمة الدالة LET قد تم حذفها من الجملة الثانية والثالثة .

٢ - ٩ قراءة المدخلات — جملة INPUT READING INPUT—THE INPUT STATEMENT

تستخدم جملة **INPUT** لإدخال بيانات رقمية أو حرفية للحاسب خلال تنفيذ البرنامج ، تحتوي الجملة على رقم الجملة ثم الكلمة الدالة **INPUT** وقائمة من المتغيرات . يمكن أن تتضمن القائمة كلا من المتغيرات الرقمية والحرفية . ويجب فصل المتغيرات عن بعضها بواسطة فاصلات (,) .

مثال ٢ - ١٤

```
5 INPUT A,B,C
10 INPUT N$,M$,X0,F5
15 INPUT P(I),Q(I),T$(I)
```

تسمى المتغيرات المبينة في الجملة الأخيرة متغيرات ذات أدلة . سوف نناقش المتغيرات ذات الأدلة في الفصل الخامس .

عند مقابلة جملة **INPUT** أثناء تنفيذ البرنامج ، تطبع علامة استفهام (؟) على الكونسول (نضد التشغيل) مشيراً إلى طلب بيانات . وعادة ما تظهر علامة الاستفهام عند ابتداء سطر جديد . يتوقف تنفيذ باقى البرنامج حتى يتم إمداد البرنامج بالبيانات المطلوبة .

يجب أن يعطى المبرمج (أو مستخدم البرنامج) المعلومات المطلوبة فور ظهور علامة الاستفهام بطباعة البيانات الصحيحة على لوحة الكونسول يتمها رجوع عربة الآلة . بعد ذلك تنقل البيانات إلى ذاكرة الحاسب ويستكمل تنفيذ البرنامج . وبذلك فإن جملة **INPUT** يمكن أن تكون مفيدة خصوصاً في أسلوب البرمجة التخطي .

يجب ملاحظة القواعد التالية عند إدخال بيانات الإدخال المطلوبة :

١ - يجب أن تناظر بنود البيانات قائمة المتغيرات في جملة **INPUT** من حيث العدد والنوع (أى يجب إعطاء الحاسب أرقاماً للمتغيرات الرقمية وحروفاً للمتغيرات الحرفية) وسوف نتجاهل أى بنود بيانات زائدة .

٢ - يجب فصل بنود البيانات عن بعضها بواسطة فاصلات (,) .

٣ - يجب أن تتكون بنود البيانات من أرقام وسلاسل حروف وغير مسموح بالصيغ الرياضية .

٤ - يجب حصر سلاسل الحروف المحتوية على فاصلات (,) أو المبتدأة بفراغات خالية بين علامات اقتباس " " . يمكن أن تحصر أى سلاسل حرفية أخرى بين علامات اقتباس حسب الرغبة .

مثال ٢ - ١٥

افرض أننا قابلنا هذه الجملة :

```
60 INPUT X,Y,C$
```

أثناء تنفيذ برنامج اليبسك ، وسوف يسبب ذلك طباعة علامة استفهام عند بداية سطر جديد على لوحة الكونسول . وسوف يتوقف تنفيذ البرنامج مؤقتاً .

وعندما يرى المستفيد علامة الاستفهام ، فسوف يشرع في إدخال البيانات المطلوبة . نفرض أن القيم الحقيقية للمتغيرات X و Y و C\$ هي 5 و 1.2×10^{-3} و "NOVEMBER 27, 1937". فيجب أن يظهر سطر بيانات الإدخال كالتالى :

"NOVEMBER 27, 1937",-1.2E-3, 5 ?

بعد طباعة البيانات ، يضغط المستفيد على مفتاح رجوع العربة وذلك يتسبب في إرسال البيانات إلى ذاكرة الحاسب . ثم يستكمل تنفيذ باقى البرنامج بعد ذلك بالطريقة العادية .

إن جملة INPUT مفيدة للغاية فى البرامج الأولية التى لا تتطلب كيات كبيرة من بيانات الإدخال . ومع ذلك ، فإن إدخال البيانات خلال جملة PRINT فيه استهلاك للوقت نسبياً ، ولا يمكن تخزين البيانات التى تم إدخالها بهذه الطريقة لاستخدام لاحق . (فى معظم أنظمة البيسك يمكن تخزين برنامج لمدة غير محددة وإعادة تشغيله كلما دعت الحاجة إلى ذلك) . سوف نرى طريقة أخرى لتعريف البيانات فى الفصل الخامس .

٢ - ١٠ طباعة المخرجات — جملة PRINT PRINTING OUTPUT—THE PRINT STATEMENT

تستخدم جملة PRINT لإرسال مخرجات بيانات من الحاسب سواء كانت رقمية أو حرفية وتتكون الجملة من رقم جملة ، ثم الكلمة الدالة PRINT وقائمة من بنود المخرجات . يمكن أن تكون بنود المخرجات أرقاماً أو صيغاً رياضية أو سلاسل حرفية . البنود المتتالية يجب أن تفصل عن بعضها بواسطة إما فصلة (و) أو فصلة منقوطة (؛) .

مثال ٢ - ١٦

مبين أدناه عدة جمل PRINT نموذجية :

```
100 PRINT A,B,C
110 PRINT "X=";X,"Y=";Y
120 PRINT "NAME:";N$,"ADDRESS:";A$
130 PRINT
140 PRINT K;C$(K);5*X0+2/2;U(I)+V(I);P$
```

تسمى المتغيرات V(I) و U(I) و C\$(K) فى الجملة الأخيرة بمتغيرات ذات أدلة . وسوف نناقش المتغيرات ذات الأدلة فى الفصل الخامس .

يجب اتباع القواعد التالية عند كتابة جملة PRINT .

المباعدة بين الكلمات فى السطر

١- تبدأ كل جملة PRINT سطرًا جديدًا للمخرجات (تم مناقشة استثناء واحد فى القواعد ٥ و ٦ أدناه) . ومع ذلك ، فسوف ينشأ سطران أو أكثر من المخرجات بواسطة جملة PRINT واحدة إذا كانت قائمة بنود البيانات فيها عدد كبير من المداخل .

مثال ٢ - ١٧

سوف تسبب جملة PRINT

```
50 PRINT C1,C2,C3,C4,C5,C6,C7,C8
```

في طباعة قيم C1 إلى C5 على سطر واحد وطباعة قيم C6 إلى C8 على السطر التالي . ونفرض مثلا أن C1 إلى C8 تمثل بالقيم التالية .

C1=8
C2=-12
C3=6.5
C4=5000
C5=0
C6=.0047
C7=-8
C8=7.2E-15

فإن المخرجات سوف تظهر كالتالي :

3	-12	6.5	5000
0.0047	-8	7.20000E-15	

٢ - إذا لم تحتو جملة PRINT على أى بنود بيانات فسوف يظهر سطر خال وهذه طريقة مفيدة للتحكم في الفراغ الرأسى لبيانات المخرجات .

مثال ٢ - ١٨

سوف تسبب جملة الطباعة التالية :

```
40 PRINT C1,C2,C3,C4
50 PRINT
60 PRINT C5,C6,C7,C8
```

طباعة القيم C1 إلى C4 على سطر واحد وطباعة القيم C5 إلى C8 على سطر آخر بينها سطر خال .

إذا كانت قيم C1 إلى C8 هى نفس القيم المشار إليها في مثال ٢ - ١٧ فإن المخرجات الناتجة من جملة PRINT الثلاث السابقة سوف تظهر كالتالي :

3	-12	6.5	5000
0	0.0047	-8	7.20000E-15

الأرقام المعنوية Significant Figures

٣ - سوف تظهر المخرجات للكميات الرقمية كالتالي :

في معظم نسخ البيسك سوف تطبع أى كمية صحيحة تحتوى على 8 خانات أو أقل كرقم صحيح أما إذا تعدى الرقم الصحيح 8 خانات فسوف يقرب إلى 6 أرقام معنوية ويطبع كرقم عشري له أس .

سوف تطبع الكمية العشرية كرقم عشري . إذا كانت الكمية تحتوى على أكثر من 6 خانات (متضمنا أى أصفار تسبق الرقم بين العلامة العشرية) فسوف تقرب إلى 6 خانات . سوف تظهر صورة أسية إذا كانت قيمة الرقم تتعدى 999999 أو إذا كانت أقل من 0.1 وتحتوى على أكثر من 6 أرقام معنوية .

مثال ٢-١٩

نفرض أن برنامج البيسك يحتوى على المتغيرات A و B و C و D و E و F إلى حددت لها القيم التالية :

```
A=1234567
B=123456789
C=-0.001234
D=0.000012345
E=-1234.5
F=1234567.89
```

فسوف تولد الجمل :

```
100 PRINT A,B,C
110 PRINT D,E,F
```

السطران التاليان من المخرجات :

```
1234567      1.23457E+8      -0.001234
1.23450E-5    -1234.5      1.23457E+6
```

سلاسل الحروف Strings

٤ - يجب أن تحصر سلاسل الحروف بين علامتى اقتباس (انظر مثال ٢-٢٠ فيما يلى) .

المباعدة بين بنود المخرجات فى السطر Spacing of Output Items Within a Line

٥ - إذا كانت بنود البيانات فى قائمة المخرجات مفصولة عن بعضها بواسطة فاصله (و) فإن كل سطر سوف يقسم إلى 5 مناطق متساوية الطول وسوف تطبع قيمة واحدة فى كل منطقة .

مثال ٢-٢٠

يحتوى برنامج بيسك على الجملة التالية :

```
65 PRINT "NAME",N$,X,5*(C1+C2)
```

فإذا كانت المتغيرات قد تم تحديثها بالقيم :

```
N$=CINNAMON      C1=7
X=89              C2=11
```

فإن جملة PRINT السابقة سوف تولد السطر التالى من المخرجات :

```
NAME      CINNAMON      39      9
```

نبين توضيحات أخرى لاستخدام الفصلة (و) فى جملة PRINT . فى الأمثلة ٢-١٧ ، ٢-١٨ ، ٢-١٩ .

إذا تلى البند الأخير من قائمة البيانات فصلة (و) فإن الكمية المخرجة التالية (أى الكمية الأولى من المخرجات فى جملة PRINT اللاحقة) سوف تطبع على نفس السطر فى حالة وجود مسافة كافية . (لاحظ أن هذه نتيجة استثناء للقاعدة 1 على الصفحة ٤٠) .

مثال ٢ - ٢١

سوف تسبب الجمل التالية :

```
100 PRINT A,B,C,
110 PRINT D,E,F
```

في طباعة قيم A و B و C و D و E على سطر واحد وتتبعها قيمة F على السطر التالي .

إذا تم تحديد المتغيرات A و B و C و D و E و F بنفس القيم العددية كما في المثال ٢ - ١٩ فالخروجيات الناتجة من جمل الطباعة السابقة سوف تظهر كالتالي :

```
1234567      1.23457E+8      +0.001234      1.23450E-5      -1234.5
1.23457E+6
```

(قارن ما سبق بالنتائج في المثال ٢ - ١٩) .

يمكن أن يصل عدد الفواصل التي تظهر إلى أربعة إذا دعت الحاجة . تأثير كل فصلة هو التحرك إلى بداية منطقة الطباعة التالية . وبذلك يمكن طباعة البيانات بفواصل واسع بينها بهذه الطريقة .

مثال ٢ - ٢٢

برنامج يبسك يحتوى على الجمل التالية :

```
120 PRINT A,B,C,D,E
180 PRINT F,,,G
```

إذا تم تحديد المتغيرات بالقيم التالية :

```
A=1      C=3      E=5      G=7
B=2      D=4      F=6
```

فإن الجمل السابقة سوف تنتج طباعة السطور التالية :

```
1          2          3          4          5
6          7
```

٦ - إذا تم استخدام الفصلة المنقوطة (;) فضلا عن الفصلة (و) وذلك لفصل بنود البيانات العددية في قائمة المخرجات فإن قيم المخرجات سوف تظهر أكثر تقارباً من بعضها . سوف تتوقف عدد المسافات المتروكة على عدد أرقام الخانات أو الحروف في كل بند من بنود المخرجات . وباستخدام الفصلة المنقوطة (;) بهذه الطريقة يمكن طباعة أكثر من 5 كيات على سطر واحد .

مثال ٢ - ٢٣

برنامج يبسك يحتوى على الجملة التالية :

```
100 PRINT A1;A2;A3;A4;A5;A6;A7;A8
```

إذا تم تحديد المتغيرات بالقيم التالية :

```
A1=11      A5=15
A2=12      A6=16
A3=13      A7=17
A4=14      A8=18
```

فإن جملة PRINT السابقة سوف تولد السطر التالى من المخرجات :

11 12 13 14 15 16 17 18

إذا تبعت سلسلة من الحروف أو متتير حرفى الفصلة المنقوطة (;) فى قائمة المخرجات ، فإن سلسلة الحروف سوف تطبع بدون ترك أى مسافات وسوف يطبع البند التالى من البيانات فوراً بعد سلسلة الحروف .

مثال ٢ - ٢٤

برنامج ببسك يحتوى على الجملة التالية :

200 PRINT "X=";X,"Y=";Y

إذا تم تحديد المتغيرات بالقيم $X = 12$ و $Y = -5$ فإن الجملة السابقة سوف تولد سطر المخرجات التالى :

X = 12 Y = -5

تسبق جملة PRINT فى معظم برامج البيسك جملة INPUT وتحتوى على سلسلة حروف . والفرض من جملة INPUT هو إصدار رسالة سريعة لطلب البيانات المطلوبة . إذا تبعت سلسلة الحروف فصلة منقوطة (;) فإن علامة الاستفهام الناتجة من جملة INPUT سوف تظهر عند نهاية الرسالة المطبوعة .

مثال ٢ - ٢٥

تم كتابة برنامج ببسك لحساب مساحة ومحيط دائرة . أول خطوة عند تنفيذ البرنامج هى قراءة قيمة نصف القطر . حيث يحتوى البرنامج على الجمل التالية :

10 PRINT "RADIUS=";
20 INPUT R

وسوف تسبب هاتان الجملتان طباعة السطر التالى من المخرجات .

RADIUS=?

وبعد ذلك يدخل المستخدم قيمة R (نصف القطر) كما هو موضح فى المثال ١ - ٥ .

وأخيراً ، يجب أن يفهم أن تأثير وضع الفصلة المنقوطة (;) بعد آخر مدخل من قائمة البيانات هو تماماً نفس التأثير عند وضع الفصلة (و) فى هذا المكان (أى أن الكمية التالية فى الطباعة سوف تظهر على نفس السطر) وقد تم توضيح ذلك فى مثال ٢ - ٢٥ .

٢ - ١١ جملة END THE END STATEMENT

تشير جملة END إلى نهاية برنامج ببسك . تحتوى الجملة ببساطة على رقم الجملة ، تتبعها الكلمة الدالة END . وهذه الجملة مطلوبة فى نسخ ببسك القديمة (وهى اختيارية فى بعض النسخ الحديثة) ، وإذا ما طلبت فلا بد أن تكون آخر جملة فى البرنامج ، كما يجب أن تحمل أعلى رقم جملة فى البرنامج .

أحد استخدامات جملة END موضح فى مثال ١ - ٦ . انظر مثال ٢ - ٢٦ لترى توضيحاً آخر لذلك .

٢ - ١٢ كتابة برامج كاملة بلغة البيسك WRITING COMPLETE BASIC PROGRAMS

تعلمنا حتى الآن كيفية قراءة بيانات للحاسب وتنفيذ حسابات رياضية وكتابة النتائج . وبذلك يمكننا القيام بكل الخطوات الهامة في برنامج بيسك كامل (ولكن بسيط) .

في الفصل الثالث ، سوف نناقش ميكانيكية إدخال برنامج للحاسب ، وتنقيح البرنامج ثم تنفيذه ، ولكن سوف نهم الآن بكتابة برامج بسيطة ويوضح مثال ٢ - ٢٦ مثل هذه البرامج . ونحن نحث القارئ لكتابة برامج قليلة أخرى خاصة به وتحمل نفس الطبيعة . (اقترحات عديدة مطاة في نهاية هذا الفصل) .

مثال ٢ - ٢٦ جذور المعادلة التربيعية

نرغب في حساب جذور المعادلة التربيعية باستخدام الصيغة الرياضية المعروفة .

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

ولنفرض أن قيم a و b و c كلها تجعل $b^2 - 4ac$ دائماً قيمة موجبة وبذلك لا نبالي عند حساب الجذر التربيعي بقيمة سالبة .

الخطوات التي يجب أن تتبع هي :

١ - قراءة القيم الرقية a و b و c

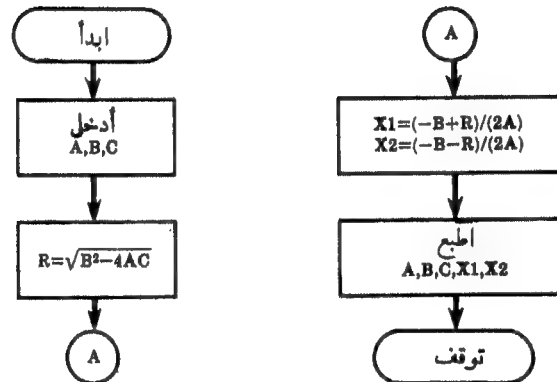
٢ - حساب قيمة $\sqrt{b^2 - 4ac}$

٣ - حساب قيم x_1 و x_2 باستخدام الصيغ الرياضية السابقة .

٤ - طباعة قيم a و b و c و x_1 و x_2

٥ - توقف .

وبين شكل ٢ - ١ خريطة سير العمليات المناظرة لذلك



شكل ٢ - ١

إذا استخدمنا المواد التي تم تقديمها سابقاً في هذا الفصل ، فكتابة برنامج كامل يقوم بهذه الحسابات يكون سهلاً للغاية . ويمكن تنفيذ الخطوة الأولى بواسطة جملة PRINT و جملة INPUT والخطوتان ٢ ، ٣ باستخدام جملة LET. وجملة PRINT تصبح مطلوبة لتنفيذ الخطوة ٤ وجملة END للخطوة ٥ . يبين شكل ٢ - ٢ برنامج ببسك المطلوب، لاحظ أن بيانات الإدخال تحتها خط .

```
10 PRINT "ENTER VALUES FOR A, B AND C"
20 INPUT A,B,C
30 LET R=(B^2-4*A*C)^.5
40 LET X1=(-B+R)/(2*A)
50 LET X2=(-B-R)/(2*A)
60 PRINT
70 PRINT "A=";A,"B=";B,"C=";C
80 PRINT "X1=";X1,"X2=";X2
90 END
```

>RUN

ENTER VALUES FOR A, B AND C
?2,5,3

A= 2 B= 5 C= 3
X1=-1 X2=-1.5

شكل ٢ - ٢

في نهاية الشكل ٢ - ٢ تظهر المخرجات المتولدة بواسطة البرنامج $a = 2$ و $b = 5$ و $c = 3$ (سوف تناقش طريقة تنفيذ برنامج ببسك في الفصل الثالث) نرى أن قيمة x_1 هي -1 وقيمة x_2 هي -1.5

٢ - ١٣ التعليقات على البرنامج — جملة REM

PROGRAM COMMENTS—THE REM STATEMENT

الطريقة الأكثر شيوعاً لتقديم ملاحظات (تعليقات) في برنامج ببسك هي استخدام جملة (REMARK) . تحتوي هذه الجملة على رقم جملة تتبعها الكلمة الدالة REM ورسالة نصية . يمكن إضافة جملة REM في أي مكان من برنامج ببسك .

مثال ٢ - ٢٧

ونبين فيما يلي جملة REM نموذجية :

```
5 REM PROGRAM TO CALCULATE THE ROOTS OF A QUADRATIC EQUATION
```

سوف نعدنا هذه الجملة بعنوان مناسب للبرنامج المعروض في مثال ٢ - ٢٦ .

لا تعطى كلمة REM الحاسب أي أمر للتنفيذ . ولكنها مع ذلك ستظهر في القائمة مع كل الجمل الأخرى الموجودة في برنامج ببسك بالتسلسل الصحيح . وبذلك فإنها تعرض على المبرمج طريقة مناسبة لتوثيق البرنامج (أي تعطى عنواناً للبرنامج لتعريف المتغيرات المهمة وتميز الأجزاء المنطقية الرئيسية من البرنامج) . وسوف نرى توضيحات كثيرة لاستخدام جملة REM في أمثلة لاحقة .

يترتب على إحاطة جملة REM بسطور خالية ظهور جملة REM بوضوح عن باقي البرنامج ، وذلك يضيف الوضوح إلى توثيق البرنامج .

ففي بعض الأحيان يتطلب إضافة تعليق يوضح جملة لها دلالة معينة في برنامج بيسك ويمكن إنجاز ذلك بالطبع باستخدام جملة REM ، ومع ذلك يمكن أن يكون وضع التعليق على نفس السطر الموجود بها الجملة طريقة أكثر تفضيلاً . يجب أن يكون التعليق من هذا النوع مسبقاً بفصله عليها (،) وذلك لتمييزها عن نهاية الجملة . مثل هذا التعليق لا يمكن أن يمتد الطول الباقي من السطر .

مثال ٢ - ٢٨

أضف التعليقين CALCULATE SECOND ROOT, CALCULATE FIRST ROOT للجملة الرابعة والخامسة من البرنامج المبين في شكل ٢ - ٢ على الترتيب .

سوف تظهر الجملتان الرابعة والخامسة كالتالي :

```
40 LET X1=(-B+R)/(2*A) 'CALCULATE FIRST ROOT
50 LET X2=(-B-R)/(2*A) 'CALCULATE SECOND ROOT
```

٢ - ١٤ تحويل التحكم — جملة GO TO

TRANSFERRING CONTROL—THE GO TO STATEMENT

عادة ما يتم تنفيذ جمل برنامج بيسك بنفس ترتيب ظهورها واحدة تلو الأخرى . ومع ذلك ، في بعض الأحيان يصبح من الضروري أن « نقفز » إلى بعض الأجزاء الأخرى من البرنامج . وبذلك نغير التسلسل الطبيعي للتنفيذ . ويمكن إنجاز ذلك بواسطة جملة GO TO . ودائماً ما نشير إلى مثل هذا القفز كعملية تفرع غير مشروط أو تحويل التحكم . وعلى ذلك فجملة GO TO تسمح لنا بتحويل التحكم لأي جملة أخرى في برنامج بيسك (بما في ذلك جملة REM) .

تحتوي جملة GO TO على رقم جملة تتبعها الكلمة الدالة GO TO ورقم السطر أو الجملة التي ينتقل إليها التحكم .

مثال ٢ - ٢٩

يحتوي برنامج بيسك على هذه الجملة التالية :

```
100 GO TO 10
```

وبذلك فإن الحاسب يؤمر بتنفيذ الجملة رقم 10 بعد ذلك .

وسوف نهم بمميزات التفرع بطريقة تفصيلية في الفصل الرابع . أما الآن فسوف نقصر اهتمامنا لتطبيق بسيط وهام لجملة GO TO .

٢ - ١٥ تكرار تنفيذ البرنامج REPETITIOUS PROGRAM EXECUTION

تتجمل مواقف كثيرة تتطلب استخدام برنامج بيسك لتشغيل عدة مجموعات من البيانات واحدة تلو الأخرى . ويمكن إنجاز ذلك ببساطة بإنهاء البرنامج بتحويل التحكم مرة ثانية لجملة « قراءة المدخلات » ، وذلك بسبب قراءة مجموعة متتالية من بيانات الإدخال للحاسب ثم تشغيلها . (لاحظ أننا نشير إلى النهاية المنطقية للبرنامج وليس النهاية المعنوية . آخر جملة معنوية في البرنامج يجب أن تكون أيضاً جملة END) . سوف نستمر في هذا الإجراء حتى يتم الانتهاء من تشغيل كل بيانات الإدخال ، عند هذا الوقت يجب على المستفيد بها أن ينهي اتصاله بالحاسب .

دائماً يتم تحويل التحكم بواسطة جملة GO TO وهذا موضح في مثال ٢ - ٣٠ .

مثال ٢ - ٣٠

دعنا نعدل في برنامج البيسك المعروض في المثال ٢ - ٢٦ وبذلك يمكننا تشغيل عدة مجموعات من بيانات الإدخال (أي أنصاف أقطار مختلفة) على التوالي . إذا فحصنا البرنامج المبين في شكل ٢ - ٢ سنرى أنه من الأسهل أن نقوم بهذه التعديلات إذا أضفنا

جملة "GO TO 10" قبل جملة END مباشرة .

يبين شكل ٢ - ٣ البرنامج المعدل . لاحظ أننا أضفنا جملة PRINT خالية وجملة GO TO 10 قرب نهاية البرنامج . والفرض من جملة PRINT الخالية هي فصل بيانات الإخراج عن رسائل المدخلات اللاحقة . ولاحظ أيضا ، أننا قد أضفنا جملة REM عند بداية البرنامج وقد تم إضافة هذه التعليقات في السطور 40 و 50 لاحظ أن المعلومات التي أدخلها المستخدم قد وضع تحتها خط .

```

5  REM PROGRAM TO CALCULATE THE ROOTS OF A QUADRATIC EQUATION
10 PRINT "ENTER VALUES FOR A, B AND C"
20 INPUT A,B,C
30 LET R=(B^2-4*A*C)^.5
40 LET X1=(-B+R)/(2*A)      'CALCULATE FIRST ROOT
50 LET X2=(-B-R)/(2*A)      'CALCULATE SECOND ROOT
60 PRINT
70 PRINT "A=";A,"B=";B,"C=";C
80 PRINT "X1=";X1,"X2=";X2
90 PRINT
100 GOTO 10
110 END

```

>RUN

ENTER VALUES FOR A, B AND C
72,6,1

A= 2 B= 6 C= 1
X1=-0.177124 X2=-2.82288

ENTER VALUES FOR A, B AND C
73,3,0

A= 3 B= 3 C= 0
X1=-9.93411E-9 X2=-1.

ENTER VALUES FOR A, B AND C
71,3,1

A= 1 B= 3 C= 1
X1=-0.381966 X2=-2.61803

شكل ٢ - ٣

نرى المخرجات الناتجة من ثلاث مجموعات مختلفة لقيم a و b و c في نهاية شكل ٢ - ٣ . (لاحظ أن القيمة التالية لـ x_1 محسوبة -9.9×10^{-9} وليست صفراً ، وهو الحل الصحيح) . ينتهى الاتصال بالحاسب بعد تشغيل ثالث مجموعة من البيانات ، بالرغم من إمكانية استمرار الإجراءات طالما رغبتنا في ذلك .

٢ - ١٦ ملاحظات ختامية CLOSING REMARKS

تعلمنا حتى الآن القدر الكافي عن البيسك والذي يمكن المستفيد من تنظيم وكتابة برامج كاملة خاصة به بالرغم من بساطتها . والفصول القادمة سوف تبين كيفية كتابة برامج أكثر تشويقاً فيها شيء من التحدي والتعقيد .

أسئلة للمراجعة

Review Questions

- ١ - ٢ اذكر طريقتين لكتابة الأرقام (الثوابت) في البيسك .
- ٢ - ٢ لخص القواعد النحوية لكتابة الأرقام .
- ٣ - ٢ اعرض مقارنة تفصيلية بين رقم مكتوب بالترميز العلمي ورقم آخر مكتوب لقيمة عشرية مرفوعة لأس في لغة البيسك .
- ٤ - ٢ ما هي سلسلة الحروف ؟ ولماذا تستخدم سلاسل الحروف هذه ؟
- ٥ - ٢ لخص القواعد النحوية لكتابة المتغيرات الرقمية والحرفية .
- ٦ - ٢ ما هي المعاملات الرياضية المستخدمة في البيسك ؟ وما هو التدرج الهرمي الطبيعي لها ؟ وما هو الترتيب الذي تتم به العمليات الحسابية بداخل مجموعة متدرجة ؟
- ٧ - ٢ ما هي الصيغة الرياضية (التمييز الرياضي) في لغة البيسك وماذا تمثل الصيغة الرياضية ؟
- ٨ - ٢ كيف يمكن تغيير التدرج الهرمي الطبيعي للعمليات الحسابية في صيغة رياضية ؟
- ٩ - ٢ اذكر مشكلة معينة يمكن أن تظهر في عمليات الأس الحسابية . اعرض سبب المشكلة ثم صف كيف يمكن أن نتجنب مثل هذه المشكلة .
- ١٠ - ٢ ما هو الغرض من جملة LET ؟
- ١١ - ٢ لخص القواعد النحوية لكتابة جملة LET ؟
- ١٢ - ٢ ناقش أوجه الشبه والاختلافات بين جملة LET والمعادلة الجبرية ؟
- ١٣ - ٢ بأي شكل تكون قواعد كتابة جملة LET أقل صرامة من بعض نسخ البيسك ؟
- ١٤ - ٢ ما هو الغرض من جملة INPUT ؟
- ١٥ - ٢ ماذا يحدث عند مقابلة جملة INPUT أثناء تنفيذ برنامج بيسك ؟
- ١٦ - ٢ لخص القواعد النحوية لكتابة جملة INPUT .
- ١٧ - ٢ اذكر ميزتين لاستخدام جملة INPUT لإدخال البيانات للحاسب .
- ١٨ - ٢ ما هو الغرض من جملة PRINT ؟
- ١٩ - ٢ لخص القواعد النحوية التي تطبق على جمل الطباعة التالية :
 - (أ) توليد ومباعدة المسافات بين سطور المخرجات .
 - (ب) كيفية ظهور الكميات العددية والعدد الأقصى للأرقام المعنوية .
 - (ج) معاملة السلاسل الحرفية .
 - (د) المباعدة بين الكميات العددية وسلاسل الحروف بداخل سطر من المخرجات .
- ٢٠ - ٢ بأي طريقة يمكن استخدام جملة PRINT مقترنة بجملة INPUT عند قراءة بيانات للحاسب ؟
- ٢١ - ٢ ما هو الغرض من جملة END ؟ ما هي القواعد التي تزامن استخدامها ؟
- ٢٢ - ٢ ما هو الغرض من جملة REM ؟ ما هي القواعد التي تحكم استخدامها ؟

- ٢-٢٣ ما هو المقصود من توثيق البرنامج ؟ كيف يمكن القيام بتوثيق برنامج في البيسك ؟
- ٢-٢٤ ما هو الغرض من جملة GO TO ؟ وكيف تكتب ؟
- ٢-٢٥ اذكر اسم جملة واحدة يجب أن توجد في كل برنامج من برامج البيسك عند استخدام إحدى نسخ بيسك القديمة وأين تظهر هذه الجملة ؟ وماذا يمكن أن يقال عن رقم الجملة الخاصة بها ؟
- ٢-٢٦ ما هي الميزة التي توجد بالتحديد عند كتابة برنامج بيسك يمكن تنفيذه عدة مرات ؟ وهل يتطلب كية هائلة من الجهد في البرمجة عند كتابة برنامج بهذه الطريقة ؟

مسائل محلولة

Solved Problems

- ٢-٢٧ عبر عن كل من الكميات التالية كأرقام يمكن تمثيلها بلغة البيسك .

الرقم بالبيسك	الكمية
7350 or 7.35E+3	7,350
-12	-12
1000000 or 1E+6	10^6
-2053180 or -2.05318E+6	$-2,053.18 \times 10^3$
0.00008291 or 8.291E-5	0.00008291
9.563E+12	9.563×10^{12}
0.16666667	1/6

- ٢-٢٨ كتبت أرقام البيسك التالية بطريقة غير صحيحة . تعرف على الأخطاء .

الرقم	الخطأ
7,104	غير مسموح بالفصلة (,)
-+4920	غير مسموح بعلامتين متتاليتين (+ , -)
2.665E+42	قيمة الأس كبيرة جداً
0.333333333333	أرقام معنوية كثيرة جداً
4.63E-0.8	لا يمكن للأس أن يحتوي على علامة عشرية

- ٢-٢٩ تمثل كل من البنود التالية سلسلة من الحروف مثلة في لغة بيسك تعرف إذا كان أي منها مكتوباً بطريقة غير صحيحة .

سلسلة حروف	الخطأ
TWENTY-SEVEN	صحيحة
2+5=7	صحيحة
ENTER ALL INPUT DATA	طويلة في بعض نسخ بيسك
75.50	صحيحة
SYMBOL IS "X"	غير مسموح بعلامات الاقتباس

٢-٣٠ يمثل كل من البنود التالية متغيراً عددياً . تعرف إذا كان أى منها مكتوباً بطريقة غير صحيحة .

المتغير	الخطأ
XR	يجب أن يكون الحرف الثانى إن وجد رقماً صحيحاً .
Q	صحيحة
C23	حروف كثيرة
8C	الحرف الأول يجب أن يكون حرفاً هجائياً وإن وجد
BIGC	حرف آخر يجب أن يكون رقماً صحيحاً .
J8	حروف كثيرة
A\$	صحيحة
	يجب أن يكون الحرف الثانى إن وجد رقماً صحيحاً .

٢-٣١ يمثل كل من البنود التالية متغيراً حرفياً . تعرف إذا كان أى منها مكتوباً بطريقة غير صحيحة .

المتغير	الخطأ
N\$	صحيحة
C	علامة الدولار غير موجودة .
Z\$3	آخر حرف يجب أن يكون علامة الدولار وبعض نسخ بيسك
Z3\$	تسمح بحرفين فقط .
E\$	يمكن أن يكون صحيحاً بالرغم من أن بعض نسخ بيسك
	تسمح بحرفين فقط .
	صحيحة .

٢-٣٢ اكتب صيغة رياضية بلغة بيسك تناظر كلا من التعبيرات الجبرية التالية .

التعبيرات الجبرية	صيغة بيسك
$3x + 5$	$3 * X + 5$
$i + j - 2$	$I + J - 2$
$x^2 + y^2$	$X \uparrow 2 + Y \uparrow 2$
$(x + y)^2$	$(X + Y) \uparrow 2$
$a/b + c/d$	$A/B + C/D$ or $(A/B) + (C/D)$
$(u + v)^{k-1}$	$(U + V) \uparrow (K - 1)$
$(4t)^{1/6}$	$(4 * T) \uparrow .16666667$ or $(4 * T) \uparrow (1/6)$

٢-٣٣ اكتب جملة LET لكل من المواقف التالية :

(أ) حدد قيمة المتغير C بالرقم 2.54 .

10 LET C=2.54

(ب) حدد قيمة المتغير X بالرقم 12 .

20 LET X=12

(ج) حدد قيمة المتغير N1 بالقيمة الممثلة بالمتغير N .

30 LET N1=N

(د) حدد قيمة المتغير A \$ بالسلسلة الحرفية JANUARY 31 .

40 LET A\$="JANUARY 31"

(أ) حدد قيمة المتغير T\$ بالسلسلة الحرفية المثلة بالمتغير S\$.
 50 LET T\$=S\$

(و) حدد قيمة المتغير F بالقيمة المثلة بالصيغة الرياضية (A↑2+B↑2+C↑2).
 60 LET F=A↑2+B↑2+C↑2

(ز) أضف 0.01 على القيمة الموجودة في المتغير C7 .
 70 LET C7=C7+.01

(ح) حدد قيمة المتغير I بالقيمة المثلة بالصيغة الرياضية (I + J) .
 80 LET I=I+J

٢ - ٣ اكتب جملة LET المتعددة لكل من المواقف التالية :

(أ) حدد قيمة المتغيرات C1 و C2 و C3 بالقيمة -37.5
 10 LET C1=C2=C3=-37.5

(ب) حدد قيمة المتغيرين P5\$ و P7\$ بالسلسلة الحرفية : ***** ERROR *****
 20 LET P5\$=P7\$="*****ERROR*****"

(ج) حدد قيمة المتغيرين M و N بالقيمة المثلة بالصيغة الرياضية (I + J)/K
 30 LET M=N=(I+J)/K

لاحظ أن جملة LET المتعددة غير متاحة في كل نسخ بيسك وكذلك يمكن حذف الكلمة الدالة LET في بعض نسخ بيسك ، أى

30 M=N=(I+J)/K

٢ - ٣ اكتب جملة LET تناظر كلا من المعادلات الجبرية التالية :

$$z = (x/y) + 3 \quad (أ)$$

10 LET Z=X/Y+3

$$z = x/(y + 3) \quad (ب)$$

20 LET Z=X/(Y+3)

$$w = (u + v)/(s + t) \quad (ج)$$

30 LET W=(U+V)/(S+T)

$$f = \left[\frac{2ab}{c+1} - \frac{t}{3(p+q)} \right]^{1/3} \quad (د)$$

40 LET F=(2*A*B/(C+1)-T/(3*(P+Q)))↑.33333333

$$r = \frac{6.8(a+b)^2/c - 7.2a/\sqrt{b+c}}{(a+c)^{1/n}} \quad (هـ)$$

50 LET R=(6.8*(A+B)↑2/C-7.2*A/(B+C)↑.5)/(A+C)↑(1/N)

٢-٣٦ فيما يلي معادلتان جبريتان معقدتان . استبدل كل معادلة بعدة معادلات بسيطة واكتب جملة LET المناظرة لها .

$$t = \left[\frac{2ab}{c+1} - \frac{r}{7(p+q)} \right]^{1/n} \quad (1)$$

$$t_1 = \frac{2ab}{c+1}$$

$$10 \text{ LET } T1=2*A*B/(C+1)$$

$$t_2 = \frac{r}{7(p+q)}$$

$$20 \text{ LET } T2=R/(7*(P+Q))$$

$$t = (t_1 - t_2)^{1/n}$$

$$30 \text{ LET } T=(T1-T2)^(1/N)$$

$$f = \frac{[6.8(a-b)^2/c - 7.2a/\sqrt{b+c}]^{1/7}}{[(c-a)^m + b^n]^{1/3}} \quad (ب)$$

$$f_1 = 6.8(a-b)^2/c$$

$$10 \text{ LET } F1=6.8*(A-B)^2/C$$

$$f_2 = 7.2a/\sqrt{b+c}$$

$$20 \text{ LET } F2=7.2*A/(B+C)^.5$$

$$f_3 = (c-a)^m + b^n$$

$$30 \text{ LET } F3=(C-A)^M+B^N$$

$$f = (f_1 - f_2)^{1/7} / f_3^{1/3}$$

$$40 \text{ LET } F=(F1-F2)^(1/7)/F3^(1/3) \text{ or}$$

$$40 \text{ LET } F=(F1-F2)^.14285714/F3^*.33333333$$

٢-٣٧ اكتب جملة مناسبة أو عدة جمل لكل من المواقف الموصوفة فيما يلي :

(أ) أدخل قيمًا عددية لكل من X1 و X2 و X3 و قيمة حرفية المتغير X\$ يجب أن تطبع كل البيانات على سطر واحد .

```
10 INPUT X1,X2,X3,X$
```

(ب) أدخل قيمًا عددية لكل من X1 و X2 و X3 على سطر واحد وقيمة حرفية المتغير X\$ على السطر التالي .

```
10 INPUT X1,X2,X3
```

```
15 INPUT X$
```

(ج) أدخل قيمًا عددية لكل من X1 و X2 على سطر واحد وقيمة عددية للمتغير X3 تتبعها قيمة حرفية المتغير X\$ على السطر التالي .

```
10 INPUT X1,X2
```

```
15 INPUT X3,X$
```

(د) اطبع قيم المتغيرات C1 و C2 و C3 و C4 و C5 كلها على سطر واحد .

```
50 PRINT C1,C2,C3,C4,C5
```

(هـ) اطبع قيم المتغيرات A1 و A2 و A3 على سطر واحد وقيم المتغيرات B1 و B2 و B3 على سطر آخر ويفصل بينهما سطر خال .

```
60 PRINT A1,A2,A3
```

```
65 PRINT
```

```
70 PRINT B1,B2,B3
```

(و) اطبع قيم المتغيرات A1 و A2 و A3 و B1 و B2 و B3 كلها في سطر واحد متقاربة من بعضها البعض بقدر المستطاع .

```
50 PRINT A1;A2;A3;A4;A5;A6
```

(ز) اطبع قيم المتغيرات X و Y و Z على سطر واحد ، اسبق كل قيمة عددية بعنوان مناسب .

```
100 PRINT "X=";X;"Y=";Y;"Z=";Z
```

```
100 PRINT "X=";X;"Y=";Y;"Z=";Z
```

أو

(ح) اطبع قيمة المتغير N\$ تلوها قيمة N ثم بعد ذلك قيمة الصيغة الرياضية $A\uparrow 2+B\uparrow 2$

120 PRINT N\$;N; $A\uparrow 2+B\uparrow 2$

or

120 PRINT N\$;N; $A\uparrow 2+B\uparrow 2$

أو

(ط) اطبع السلسلتين الحرفيتين LEFT و RIGHT قرب الحافة اليمنى والحافة اليسرى .

150 PRINT "LEFT",,,, "RIGHT"

(ي) اطبع الرسالة ROOTS OF SIMULTANEOUS EQUATIONS على سطر واحد في الوسط بقدر المستطاع .

200 PRINT, "ROOTS OF SIMULTANEOUS EQUATIONS"

(ك) اطلب رسالة تشير إلى طلب قيمة عددية للمتغير C ، ثم بعد ذلك أدخل القيمة العددية للمتغير C .

5 PRINT "C=";

10 INPUT C

٢-٣٨ بين كيف تظهر البيانات المدخلة في كل من المواقف التالية :

10 INPUT X1,X2,X3,X\$

(أ)

$X1=4.83 \times 10^{-3}$

$X3=941.55$

$X2=-537$

$X\$=BUCS$

حيث

?4.83E-3,-537,941.55,BUCS

أو

?00483,-537,941.55,BUCS

10 INPUT X\$,X1

(ب)

20 INPUT X2,X3

حيث تكون المتغيرات لها نفس القيم الموجودة في الجزء (أ)

?BUCS,4.83E-3

?-537,941.55

30 INPUT A,A\$,A1

(ج)

A=350

A1=-8.05

حيث

A\$=APRIL 12, 1969

?350,"APRIL 12, 1969",-8.05

٢-٣٩ بين كيف تظهر المخرجات المطبوعة في كل من المواقف التالية

10 PRINT "NAME ",N\$,(X+Y) $\uparrow 2/3$,T4

(أ)

N\$=GEORGE

Y=8.2

حيث

X=27.6

T4=-5.83 $\times 10^{-4}$

NAME

GEORGE

427.213

-0.000583

10 PRINT "NAME ";N\$;(X+Y) $\uparrow 2/3$;T4

(ب)

حيث تكون المتغيرات لها نفس القيم الموجودة في الجزء (ا)

NAME GEORGE 427.213 -0.000583

12 PRINT A1,A2,A3,A4
14 PRINT B1,B2,B3,B4 (ج)

A1=7.43 × 10³ B1=-2.55 × 10⁻⁸ حيث
A2=-4373665.8 B2=0.843 × 10⁷
A3=0.0006066183 B3=400.33
A4=-3136687 B4=10⁻⁸

7430 -4.37367E+6 6.06618E-4 -3136687
-2.55000E-8 8430000 400.33 0.001

12 PRINT A1;A2;A3;A4;
14 PRINT B1;B2;B3;B4 (د)

حيث تكون المتغيرات لها نفس القيم الموجودة في الجزء (ج)

7430 -4.37367E+6 6.06618E-4 -3136687 -2.55000E-8 8430000 400.33
0.001

٢-٤٠ بين كيف يمكن وضع التعليق (الملاحظات) في برنامج بيسك في كل من المجالات التالية :

(ا) أضف عنوان البرنامج AREA AND CIRCUMFERENCE OF A CIRCLE

10 REM AREA AND CIRCUMFERENCE OF A CIRCLE

(ب) أضف التعليقات AREA و CIRCUMFERENCE للجميل التالية :

40 LET A=P*R^2

50 LET C=2*P*R

40 LET A=P*R^2 'AREA
50 LET C=2*P*R 'CIRCUMFERENCE

٢-٤١ عدة جمل GO TO مبينة فيما يلي . تعرف إذا كان أي منها مكتوباً بطريقة غير صحيحة .

الخطأ	الجملة
صحيحة	10 GO TO 50
رقم الجملة التي يتحول إليها التحكم يجب أن يكون رقماً صحيحاً موجباً وليس متغيراً .	120 GO TO M
صحيحة	80 GO TO 25
لا يمكن جملة GO TO أن تحول التحكم إلى نفسها .	50 GO TO 50

مسائل تكميلية Supplementary Problems

٢-٤٢ أجب عن الأسئلة التالية بما هو متاح في نسخ بيسك المستخدمة في مدرستك أو مكتبك .

- (أ) كم عدد الأرقام المعنوية التي يمكن أن يتضمنها أى عدد ؟
 (ب) كم عدد الحروف التي يمكن أن تظهر في سلسلة حرفية (أى ما هو أقصى طول للسلسلة الحرفية) ؟
 (ج) هل يمكن أن يكتب متغير حرفي كحرف يتبعه رقم ثم يتبعه علامة الدولار ؟ (مثلا C1\$)
 (د) هل يمكن حذف الكلمة الدالة LET من جملة LET (مثلا 10 A = B + C)
 (هـ) هل مسيوح بتحديد قيم لأكثر من متغير في جملة LET واحدة (مثلا 10 LET X = Y = Z = 13) ؟

٢-٤٣ عبر عن كل من الكميات التالية كمعد من أعداد بيسك .

-7,328,500	(أ)	5	(ا)
0.2851×10^4	(و)	8000	(ب)
0.2851×10^{10}	(ز)	-1.8033×10^{-9}	(ج)
-16,752.47	(ح)	1/8	(د)

٢-٤٤ بعض الثوابت التالية كتبت بطريقة خاطئة . تعرف على كل الأخطاء .

0.8333333333333333E-2	(ط)	-77777777	(أ)	+0.250	(ا)
-00263	(ى)	1,000,000	(و)	5076	(ب)
4.48E	(ك)	2.53E+99	(ز)	3 E-2	(ج)
0.83333333-E2	(ل)	64E+6	(ح)	3.8822E-7.3	(د)

٢-٤٥ بعض السلاسل الحرفية التالية كتبت بطريقة خاطئة . تعرف على كل الأخطاء .

\$1,995.00	(ا)
JULY 4, 1776	(ب)
BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE	(ج)
4 O'CLOCK	(د)
"NUTS!"	(هـ)
2X+4Y=Z	(و)

٢-٤٦ يمثل كل متغير من المتغيرات التالية إما متغير رقمي أو متغير حرفي ولكن بعضها مكتوب بطريقة خاطئة . حدد نوع المتغير في كل حالة صحيحة ثم تعرف على كل الأخطاء .

Z0 (م)	5T (ك)	PI (ط)	XSTAR (ز)	\$J6 (أ)	J\$ (ج)	J (ا)
M 5 (ن)	Y* (ل)	N\$ (ى)	C10 (ح)	J6\$ (و)	J\$6 (د)	J6 (ب)

٢-٤٧ أكتب صيغة رياضية تناظر كلا من الحدود الجبرية التالية :

$\frac{2(p/q)^{k-1}}{(r-3t)^{1/m}}$	(و)	t^{n+1}	(ا)
$(i+j-1)^{2/5}$	(ز)	$(x+3)^{1/k}$	(ب)
$\left[\frac{(x_1+x_2)^m(y_1+y_2)^n}{(x_1/y_1)^{m+n}(x_2/y_2)^{m-n}} \right]^{1/mn}$	(ح)	$2(a/b)^{1/3}$	(ج)
		$1.87(u+v) - 5.088(x/y + 2z^2)$	(د)
		$1 - x + x^2/2 - x^3/6 + x^4/24 - x^5/120$	(هـ)

٢-٤٨ في نسخة اليسك التي لديك ، هل يمكن لتغير لم يتم تعريفه (أى لم تخصص له قيمة) أن يظهر على الطرف الأيمن من علامة التساوى في جملة LET ؟ وإذا ظهر فما هو تأثير ذلك ؟

٢-٤٩ اكتب جملة LET لكل من المواقف التالية :

- (أ) حدد قيمة المتغير P بالرقم 758.33
- (ب) حدد قيمة المتغير B بالقيمة المثلثة بالمتغير A .
- (ج) حدد قيمة المتغير F\$ بالقيمة PITTSBURGH , PA
- (د) حدد قيمة المتغير N\$ بالقيمة المثلثة في المتغير M\$
- (هـ) حدد قيمة المتغير Y3 بالصيغة الرياضية $X / (A + B + C)$
- (و) انقص 2 من القيمة الموجودة في المتغير K .
- (ز) ضاعف القيمة الموجودة في المتغير C5 .
- (ح) حدد قيمتي المتغيرين B و C بقيمة الصيغة الرياضية : $(A+2+B+2)^{+.5}$

٢-٥٠ اكتب جملة LET التي تناظر كلا من المعادلات الجبرية التالية :

$$w = \frac{(a+3)b^n}{2.7(c-d/b) + 1} \quad (أ)$$

$$f = \left\{ \frac{(a/b)^n / (c-d)^m}{[d/(b-a)^{n+m}]} \right\}^{1/(n+m)} \quad (ب)$$

$$y = \frac{a_1 - a_2x + a_3x^2 - a_4x^3 + a_5x^4}{c_1 - c_2x + c_3x^2 - c_4x^3} \quad (ج)$$

$$P = rA(1+r)^n / [(1+r)^n - 1] \quad (د)$$

٢-٥١ سوف تنتج كل من المعادلات المذكورة في مسألة ٢-٥٠ جملة LET طويلة . استغنى عن كل جملة ببضع جمل قصيرة متتالية من جمل LET البسيطة .

٢-٥٢ اكتب معادلة جبرية تناظر كلام من جمل LET التالية :

$$10 \text{ LET } F=A+2*B/C^{+.5} \quad (أ)$$

$$20 \text{ LET } F=A+(2*B/C)^{+.5} \quad (ب)$$

$$30 \text{ LET } F=(A+2)*(B/C)^{+.5} \quad (ج)$$

$$40 \text{ LET } F=((A+2)*B/C)^{+.5} \quad (د)$$

$$50 \text{ LET } G=P*Q/R*S/T \quad (هـ)$$

٢-٥٣ ما هي الصعوبة التي يمكن أن نواجهها عند تنفيذ الجملة :

$$15 \text{ LET } X=(Y-Z)^{+.25}$$

٢-٥٤ ادرس الجملة التالية :

$$25 \text{ LET } P=-Q^{+4}$$

ما هي القيمة التي تعطى للمتغير P إذا كانت قيمة Q = 2 ؟

٥٥-٢ ادرس الجملة التالية :

35 LET P=Q+4

ما هي القيمة التي تعطى للمتغير P إذا كانت قيمة $Q = -2$ ؟ (قارن النتيجة بإجابة المسألة ٥٤-٢ السابقة).

٥٦-٢ اكتب جملة مناسبة أو مجموعة من الجمل لكل من المواقف الموصوفة التالية :

- (أ) أدخل قيمًا عددية للمتغيرات A و B و C وقيمًا حرفية للمتغيرات M\$ و N\$. يجب أن تطبع كل البيانات على سطر واحد من سطور النهاية الطرفية.
- (ب) أدخل القيم الممنوعة للمتغيرات A و N\$ و B على سطر واحد والقيم الممنوعة للمتغيرات M\$ و C على السطر التالي.
- (ج) أدخل القيم العددية للمتغيرات A و B و C والقيم الحرفية للمتغيرات M\$ و N\$. يجب أن تطبع كل قيمة من هذه القيم في بداية سطر جديد.
- (د) اطبع رسالة تقول :

ENTER VALUES FOR A,B,C,M\$ AND N\$

- ثم أدخل بعد ذلك البيانات المطلوبة على نفس السطر المطبوع عليه الرسالة.
- (هـ) اطبع الرسالة الممنوعة في الجزء (د). ثم بعد ذلك أدخل البيانات المطلوبة على السطر التالي.
- (و) اطبع قيم المتغيرات A و B و C و M\$ و N\$ على سطر واحد مع الفاصل بينهما بمسافات طبيعية.
- (ز) اطبع قيم المتغيرات A و B و C على سطر واحد ، اجمل المسافات بين البنود متقاربة بقدر المستطاع . واسمح لمخرجات متعاقبة أن تبدأ على نفس السطر فوراً بعد قيمة المتغير C .
- (ح) اطبع قيم المتغيرات A و B و C و $(A+B+C)/3$ و $(A*B*C)^{(1/3)}$ و $(A^2+B^2+C^2)^{.5}$ وكلها على سطر واحد واتبعها بسطر خال ثم سطر ثالث تطبع عليه قيمة M\$ قرب الهامش الأيسر وقيمة N\$ قرب الهامش الأيمن .
- (ط) اطبع القيم العددية للمتغيرات A و B و C وكلها على سطر واحد . ويسبق كل عدد عنوان مناسب يصف العدد .
- (ي) اطبع قيم المتغيرات M\$ ، N\$ على سطرين منفصلين يفصل بينهما سطر خال . اسبق قيمة المتغير M\$ بالعنوان NAME واسبق قيمة المتغير N\$ بالعنوان SOCIAL SECURITY NUMBER حاول أن تجعل المخرجات في المنتصف ومتقاربة من بعضها بقدر المستطاع .

٥٧-٢ بين كيف تظهر البيانات المدخلة لكل من المواقف التالية :

- (أ)
- ```
5 INPUT A,B,C
10 INPUT M$,N$
```
- A = 0.0000062      M\$ = SHARON      حيث  
B =  $27.5 \times 10^{-12}$       N\$ = GAIL  
C = -1000
- (ب)
- ```
20 INPUT P1,P2,T$
```
- P1 = -743.08 T\$ = SUSAN حيث
P2 = 0.00987
- (ج)
- ```
25 INPUT A$,B$,C$
```
- A\$ = NEW YORK      C\$ = SAN FRANCISCO      حيث  
B\$ = CHICAGO
- (د)
- ```
15 INPUT P,P$,Q,Q$
```
- P = 2,770,543 Q = 48.8×10^9 حيث
P\$ = DECEMBER 29, 1963 Q\$ = ELEVEN O'CLOCK

٥٨-٢ بين كيف تظهر البيانات المخرجة لكل من المواقع التالية :

100 PRINT A;B;C;P;P1;P2;Q

(أ)

A = 0.0000062
B = 27.5×10^{-12}
C = -1000
P = 2,770,543

P1 = -743.08
P2 = 0.00987
Q = 48.8×10^9

حيث

110 PRINT A,B,C,P,P1,P2,Q

(ب)

حيث قيم المتغيرات هي نفس القيم في الجزء (أ)

120 PRINT A+B*C,P/Q,P1/P2

(ج)

حيث قيم المتغيرات هي نفس القيم في الجزء (أ)

130 PRINT M\$,P\$,Q\$

(د)

M\$ = SHARON
P\$ = DECEMBER 29, 1963
Q\$ = ELEVEN O'CLOCK

حيث

٥٩-٢ بين كيف توضع التعليقات (أو الملاحظات) في برنامج يبسك لكل من الحالات التالية :

(أ) أضف عنوان البرنامج AVERAGING OF AIR POLLUTION DATA

(ب) أدخل الملاحظة BEGIN LOOP TO CALCULATE CUMULATIVE SUM

(ج) أضف التعليق CALCULATE AVERAGE VALUE إلى الجملة :

80 LET A=S/N

(د) أضف التعليق READ A DATA POINT إلى الجملة التالية :

20 INPUT X,T

٦٠-٢ فيما يلي نبيين عدة جمل GO TO تعرف على أى جملة كتبت خطأ .

55 GO TO 400
20 GO TO "60"

(د)
(هـ)

100 GO TO 12
75 GO TO K+1
30 GO TO 30

(أ)
(ب)
(ج)

مسائل للبرمجة

Programming Problems

٦١-٢ جهز خريطة سير عمليات البرنامج المبين في مثال ٢-٣٠ قارن بين هذه الخريطة وخريطة سير العمليات للمثال ٢-٢٦ المبينة في شكل ٢-١ .

٦٢-٢ اكتب برنامجاً كاملاً مكتوباً بلغة البيسك لكل من المسائل التالية :

(أ) اطبع HELLO في منتصف السطر .

(ب) اجعل الحاسب يطبع ؟

HI, WHAT'S YOUR NAME?

على سطر . ثم بعد ذلك يدخل المستفيد اسمه أو اسمها فوراً بعد علامة الاستفهام . ثم يترك الحاسب سطرين ويعطيه .

WELCOME (name)!

LET'S BE FRIENDS!

على سطرين متعاقبين .

٢ - ٦٣ اكتب برنامجاً كاملاً من النوع التخطي بلغة البيسك لكل من المسائل التالية :

(١) يجب أن تقرأ درجات الحرارة للحاسب بالدرجات الفهرنيتيه ثم تحول إلى درجات مئوية باستخدام الصيغة :

$$^{\circ}\text{C} = \frac{5}{9}(^{\circ}\text{F} - 32)$$

(ب) تحتوي حصالة على عدد n_1 من أنصاف الدولار و n_2 من أرباع الدولار و n_3 من وحدات العشر سنتات و n_4 من وحدات السنتات الخمسة ، و n_5 من السنتات . كم عدد النقود في الحصالة بدلالة الدولارات ؟ (الدولار مائة سنت) .

٢ - ٦٤ ابتدع مخططاً تمهيدياً ثم ارسم خريطة سير العمليات ثم بعد ذلك اكتب برنامجاً كاملاً بلغة بييسك لكل من المسائل التالية . اكتب كل برنامج بطريقة تمكنه من تشغيل عدة مجموعات من البيانات المتتالية . وتأكد من أن كل البيانات المخرجة معنونة بوضوح .

(١) احسب حجم ومساحة الكرة باستخدام الصيغ الرياضية :

$$V = 4\pi r^3/3$$

$$A = 4\pi r^2$$

حيث r هي نصف قطر الكرة .

(ب) يرتبط ضغط وحجم ودرجة الحرارة لكتلة من الهواء بالصيغة الرياضية .

$$PV = 0.37 m(T + 460)$$

حيث P = الضغط بالرطل لكل بوصة مربعة .

V = الحجم ، بالقدم المكعب ،

m = كتلة الهواء ، بالرطل ،

T = درجة الحرارة بالفهرنيت . $^{\circ}\text{F}$.

إذا كان إطار العربة يحتوي على 2 قدم مكعب من الهواء منفوخ بضغط 28 رطل لكل بوصة مربعة عند درجة حرارة الحجرة . فكم هي كمية الهواء الموجودة بالإطار ؟

(ج) إذا كانت a و b و c تمثل ثلاثة أضلاع المثلث ، فإن مساحة المثلث :

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

حيث $s = (a + b + c)/2$ وأيضاً نصف قطر الدائرة الخارجية :

$$r_i = A/s$$

ونصف قطر الدائرة الداخلية :

$$r_e = abc/(4A)$$

فاحسب مساحة المثلث ، ومساحة الدائرة الخارجية ، ومساحة الدائرة الداخلية لكل من مجموعات البيانات التالية :

a	11.88	5.55	10.00	13.75	12.00	20.42	7.17	173.67
b	8.06	4.54	10.00	9.89	8.00	27.24	2.97	87.38
c	12.75	7.56	10.00	11.42	12.00	31.59	6.66	139.01

(د) تفرض أن كمية P من الدولارات قد تم استثمارها بنسبة ربح سنوية مقدارها i (معبّر عنها بقيمة عشرية) . إذا أعيد استثمار الربح بعد عدد n من السنوات فإن إجمالي كمية النقود F يمكن تحديدها بالمعادلة $F = P(1 + i)^n$ (ويعرف هذا بقانون الربح المركب) .

- إذا استثمرت الكمية \$5000 بربح مركب سنوياً مقداره 6% فكم هي كمية النقود التي يتم تجميعها بعد عشر سنوات ؟

- إذا كان الربح المركب ربع سنوياً وليس سنوياً فإن المعادلة يجب تغييرها لتصبح $F = P(1 + i/4)^{4n}$

(هـ) زيادة عدد البكتريا المستنبطة مع الوقت تتناسب مباشرة مع عددها . وعلى ذلك كلما كبر العدد فإن البكتريا سوف يزيد عددها أسرع . يمكن التعبير عن العدد رياضياً كما يلي :

$$P = P_0[1 + 0.0289t + (0.0289t)^2/2 + (0.0289t)^3/6 + (0.0289t)^4/24 + \dots + (0.0289t)^n/n!]$$

حيث t = الوقت بالساعات بعد وقت المرجع

P_0 = عدد البكتريا عند وقع المرجع

p = عدد البكتريا عند الوقت t

احسب عامل تضاعف العدد (p/p_0) بعد 2 و 5 و 10 و 20 و 50 ساعة من وقت المرجع . اذكر الحدود العشرة الأولى من المتوالية (أى اجعل $n = 9$)

الفصل ٣

تشغيل برنامج بيسك

Running a BASIC Program

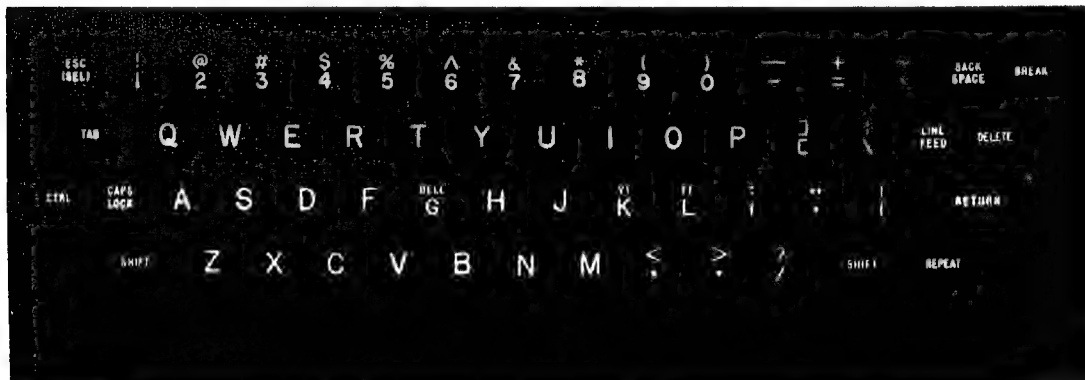
والآن وقد تعلمنا كيفية كتابة برامج بيسك بسيطة ، دعنا نرى كيفية إدراج هذه البرامج للحاسب ، وتنفيذها ، والاحتفاظ بها ، وعمل قائمة بها وتنفيذها . سوف تأخذ في الاعتبار بعض الطرق لاكتشاف بعض أنواع الأخطاء المتعددة التي يمكن حدوثها في البرامج المكتوبة بطريقة غير صحيحة وكيفية تصحيحها .

وسوف نوجه انتباهنا في هذا الفصل تجاه استخدام حاسبات ضخمة في بيئة مشاركة الزمنية . وسنفترض أن الوحدة الطرفية لإنتاج « نسخة مادية » مشابهة لتلك الموضحة في شكل ٣-١ ، تستخدم للإتصال مع الحاسب المضيف (أى الحاسب الكبير) . واستخدام نهايات طرفية من نوع Video مماثلاً لذلك ، فيما عدا أن المعلومات القديمة ستلف من أعلى الشاشة عندما تتولد المعلومات الجديدة في أسفل الشاشة .^١ وتشغيل برنامج البيسك على الحاسب الدقيق يماثل ذلك ، إلا أن التفاصيل تختلف نوعاً ما (في الحقيقة نجد أن طرق تشغيل البيسك مختلفة نوعاً ما على كل الحاسبات بصرف النظر عن حجمها) . ويحتوى الفصل العاشر (انظر قسم ١٠-٢ ، ١٠-٧) على أمثلة توضح كيف يمكن وضع برامج البيسك في قائمة وتنفيذها ، وتخزينها ، وتنفيذها على حاسب دقيق .

٣ - ١ النهاية الطرفية للمشاركة الزمنية THE TIMESHARING TERMINAL

لقد تعلمنا أنه يجب على المبرمج أن يتعامل مع الحاسب من خلال نهاية طرفية للمشاركة الزمنية (أو كونسول) وذلك عند تشغيلها في بيئة مشاركة زمنية . يمكن أن تتصل هذه الأجهزة مباشرة بواسطة سلك إلى الحاسب أو يمكن استخدامها للإتصال عن طريق خط تليفون . وفي الحقيقة فإن بعض نهايات المشاركة الزمنية تتضمن مودم مبنى داخله وقرص تليفون . وعلاوة على ذلك ، فإن بعض النهايات الطرفية يمكنها القيام بتخزين البرامج وتنفيذها وتحويلها من وإلى الحاسب المضيف . مثل هذه الأجهزة غالباً ما يطلق عليها نهايات طرفية ذكية (ذكية) .

يبين شكل ٣ - ١ صورة قريبة للوحة المفاتيح للنهاية الطرفية الممثلة في شكل ٣ - ١ . يوجد بعض النهايات الطرفية لها لوحات مفاتيح مماثلة لذلك تماماً إن لم تكن طبق الأصل . وسوف نرجع إلى بعض هذه المفاتيح في أقسام متعاقبة من هذا الفصل .



شكل ٣ - ١

٣ - ٢ التسجيل لدخول النظام LOGGING IN

الخطوة الأولى للاتصال بالحاسب من خلال نهاية طرفية في المشاركة الزمنية هو إنشاء اتصال بالحاسب . ويتم إنجاز ذلك بواسطة الإجراء Login (أو Logon) . ويمثل الوصف التالي إجراء التسجيل لدخول النظام ، بالرغم من احتمال بعض تغييرات من نسخة بيسك لأخرى (*) .

إذا كان الاتصال سوف يتم من خلال خط تليفوني ، فإن إجراء التسجيل لدخول النظام سوف يبدأ بأن يطلب المستفيد رقم تليفون معين (وذلك بعد توصيل النهاية الطرفية بمنبع الكهرباء) . ثم بعد ذلك يرد الحاسب ويطبع رسالة موجزة يطلب فيها من المستفيد الذي يريد الدخول للنظام أن يصدر login (أى يعرف نفسه وذلك بتحديد رقم مشروع . . . الخ) . ويجب على المستفيد إذن أن يطبع LOGIN (أو LOGON أو ببساطة LOG) ويتبعها بالمعلومات المطلوبة . ويمكن أن يكون من الضروري أن يطبع المستفيد BASIC ، مشيراً بذلك أنه يرغب في العمل بلغة بيسك . وليس بأى لغة أخرى .

ويكون الإجراء مائلاً ، أو يمكن أبسط ، إذا كانت النهاية الطرفية موصلة بواسطة سلك بطريقة مباشرة للحاسب ، وبذلك لا يتطلب اتصالاً بخط تليفوني وفي هذه الحالة عادة يبدأ الإجراء عندما يطبع المستفيد LOGIN يتبعها رقم المشروع الخاص به .

مثال ٣ - ١

يرغب طالب في جامعة كبيرة في تشغيل برنامج بيسك من نهاية طرفية فيها قرص تليفوني . وقد حدد المستفيد رقم المشروع 123456 وإجراء login مبين فيما بعد .

١ - يبدأ الطالب في توصيل النهاية الطرفية بالكهرباء (يوجد مفتاح التشغيل على النهاية الطرفية المبينة في الشكل ١ - ٣ قرب الناحية اليسرى من لوحة المفاتيح) .

٢ - ثم يدير الطالب قرص التليفون ويطلب الرقم الصحيح ويحجب الحاسب بطباعة

PLEASE LOGIN.

٣ - يطبع الطالب كلمة LOG بعد النقطة ثم يضغط على مفتاح RETURN ثم بعد ذلك يحجب الحاسب بطباعة

JOB 27 TTY42
#

٤ - ثم يدخل الطالب بعد ذلك رقم المشروع الخاص به ، 123456 وذلك بعد علامة // ثم يضغط على مفتاح RETURN مرة أخرى . بعد ذلك يطبع الحاسب :

08-FEB

THUR 21:11:46

مشيراً إلى التاريخ واليوم والوقت بالترتيب .

٥ - ثم تطبع الكلمة (BASIC) بعد ذلك بواسطة الطالب بعد النقطة . وتسبب هذه الكلمة التوصل إلى نظام بيسك من مكتبة الحاسب للغات البرمجة .

٦ - يطبع الحاسب :

NEW OR OLD -->

ويجب على الطالب أن يجيب طبقاً لذلك . فإذا رغب الطالب في التوصل إلى برنامج بيسك قد تم تخزينه في مكتبة الحاسب فيجب أن يطبع OLD . إلا أن الطالب يرغب في هذه الحالة إدخال برنامج جديد . ولذلك فيجب أن يطبع كلمة NEW بعد المهيم .

(*) الإجراء الموصوف في هذا الفصل ينطبق بالتحديد على نظام الحاسب DEC system-10 كما هو منفذ حالياً في جامعة بيتسبرج بالولايات المتحدة الأمريكية وللسهولة تم حذف بعض التفاصيل من النص .

٧ - وأخيراً ، يطبع الحاسب :

NEW FILE NAME-->

PLEASE LOGIN.

.LOG
JOB 27 TTY42
#123456
08-FEB THUR 21:11:46

ويجب الطالب بطباعة اسم البرنامج - في هذه الحالة SAMPLE .
(ويتكون الاسم النموذجي من عدد من الحروف يصل إلى ستة
حروف تبدأ بحرف هجائي) ويكون الطالب الآن مستعداً للبدء في
طباعة البرنامج .

.BASIC

NEW OR OLD-->NEW
NEW FILE NAME-->SAMPLE

والشكل ٣ - ٢ يبين قائمة كاملة بإجراء التسجيل لدخول النظام والبنود
التي أتم الطالب إدخالها موضوع تحتها خط .

شكل ٣ - ٢

إذا حدث خطأ في انتقال المعلومات للحاسب أثناء إجراء التسجيل لدخول النظام فسوف تطبع رسالة مناسبة ويطلب من المستخدم إعادة
إدخال المعلومات مرة ثانية .

مثال ٣ - ٢

بالإشارة للموقف الذي تم وصفه في المثال ٣ - ١ ، فلنفترض أن الطالب قد طبع BASIV بدلا من BASIC أثناء إجراء التسجيل
لدخول النظام (المفتاح V بجانب المفتاح C في أغلب لوحات المفاتيح) فإن الحاسب سوف يجب بطباعة

?BASIV

ويجب على الطالب أن يطبع BASIC بعد النقطة ويستكمل إجراء التسجيل لدخول النظام . إجراء التسجيل لدخول النظام معين كاملا
في الشكل ٣ - ٣ . مرة ثانية إجابات الطالب موضوع تحتها خط .

PITT DEC-1055/A 54A.31 21:11:46

PLEASE LOGIN OR ATTACH.

.LOG
JOB 27 PITT DEC-1055/A 54A.31 TTY42
#113421/160531
PASSWORD:
ALLOCATION REMAINING: 9.8 UNITS
2111 08-FEB THUR

.R BASIC

NEW OR OLD-->NEW
NEW FILE NAME-->EX3.1

PLEASE LOGIN.

.LOG
JOB 32 TTY42
#123456
08-FEB THUR 21:37:07

.BASIV

?BASIV

.BASIC

NEW OR OLD-->NEW
NEW FILE NAME-->SAMPLE

شكل ٣ - ٤

شكل ٣ - ٣

تصدر عديد من أنظمة المشاركة الزمنية كلمة سر منفصلة لكل مستفيد ويجب أن تعطى مع رقم المشروع . ويتم إدخال كلمة السر
وذلك بكتابتها على لوحة المفاتيح ولكن الحاسب يجعلها غير مرئية وذلك للاحتفاظ بالسرية ولا يسمح للمستخدم بالدخول للنظام إذا
لم يعط كلمة السر الصحيحة لرقم مشروع معين .

والشكل ٣ - ٤ : يبين قائمة بإجراء التسجيل لدخول النظام يتطلب كلمة السر . وكما سبق أن ذكرنا سيوضع خط تحت إجابات المستخدم .

٣ - ٣ إدخال برنامج ENTERING A PROGRAM

فور الانتهاء من إجراء التسجيل لدخول النظام يمكن للمستخدم أن يشرع في كتابة البرنامج الخاص به ، جملة واحدة (سطر واحد) في كل مرة . ولعمل ذلك يجب على المستخدم أن ينتظر حتى يطبع الحاسب الرمز

في بداية السطر . (ويعرف ذلك باللقين) . بعد ذلك يطبع المستخدم جملة ببسك مبتدئاً برقم الجملة (رقم السطر) . لن ترسل المعلومات المطبوعة للحاسب ، في هذه الحالة ، حتى يتم الضغط على مفتاح RETURN . وفور عمل ذلك يتحرك رأس الطباعة لبداية السطر التالي ، ويطلب الحاسب الجملة التالية بطباعة الرمز > مرة أخرى .

يستكمل هذا الإجراء حتى يتم الانتهاء من إدخال جميع جمل البرنامج وإرسالها للحاسب وعندما يولد الحاسب الرمز > بعد آخر جملة - فعل المستخدم أن يجيب بأمر يوضح فيه ماذا يجب عمله بالبرنامج (مثل RUN وLIST وSAVE ... الخ) . وسوف نناقش هذه الأوامر فيما بعد في هذا الفصل .

عند كتابة البرنامج للحاسب لا يستلزم أن تدخل التعليمات للحاسب بنفس الترتيب الذي يتم بها التنفيذ . فسوف يعيد الحاسب تنظيم هذه التعليمات وذلك تبعاً لتزايد أرقام الجمل فور الانتهاء من إدخال البرنامج كاملاً . (تذكر أن ترتيب أرقام الجمل هو الذي يحدد التوالى الذي يتم بناء عليه تنفيذ الجمل) .

مثال ٣ - ٣

يرغب المبرمج الذي أتم إجراء التسجيل لدخول النظام أن يدخل برنامجاً للحاسب . وبسبب السرعة نسي كتابة الجملتين الأولىين . واكتشف المبرمج خطأه قبل الانتهاء من إدخال البرنامج ، ومع ذلك ، فقد قام بكتابة الجمل المنسية . ذلك مسموح به تماماً ، حيث أن البرنامج سوف يعاد ترتيبه بالترتيب الصحيح بداخل ذاكرة الحاسب .

يبين شكل ٣ - ٥ قائمة بالجمل بنفس الترتيب الذي تم بها إدخال البرنامج .

```
> 20 PRINT "RADIUS=";
> 30 INPUT R
> 40 LET A=P*R*2
> 50 LET C=2*P*R
> 60 PRINT "R=";R,"A=";A,"C=";C
> 70 GO TO 20
> 80 END
> 5 REM PROGRAM TO CALCULATE AREA AND CIRCUMFERENCE OF A CIRCLE
> 10 LET P=3.1415927
>
```

٣ - ٤ تصحيح الأخطاء CORRECTING ERRORS

من غير المعقول عموماً أن يكتب المبرمج برنامجاً كاملاً ويدخله للحاسب بدون عمل أى خطأ عفوياً . ولذلك يجب أن نعرف طريقة تصحيح الأخطاء المكتوبة أو إضافة أو حذف أو تغيير جملة فور الانتهاء من إرسالها للحاسب . في هذا القسم سوف نرى أنه من السهل إنجاز مثل هذه العمليات في البيسك .

يمكن حذف الحروف التي طبعت خطأ بالضغط على مفتاح DELETE (على بعض النماذج الطرفية مفتاح RUBOUT) . سوف يلغى الحرف الأحدث عند الضغط على مفتاح DELETE مرة واحدة ، ويسبب الضغط على المفتاح مرتين حذف الحرفين الأكثر حداثة . . . وهكذا . ومع ذلك ، فمثل هذا الحذف يجب أن يكون قبل إرسال السطر الذي يحتوي على هذه الأخطاء للحاسب (أى قبل الضغط على مفتاح RETURN) . وبعد إلغاء الحروف التي ترغب في حذفها يمكن للمبرمج أن يكمل كتابة الحروف الصحيحة .

مثال ٣ - ٤

يرغب المبرمج في إدخال برنامج البيسك المين في شكل ٣ - ٥ للحاسب . ومع ذلك ، فبينما هو يكتب أول جملة من البرنامج فقد كتب U بدون قصد بدلا من I وبذلك ظهر السطر كما يلي :

```
>20 PRU
```

وعند اكتشافه الخطأ ضغط فوراً على مفتاح DELETE مرة واحدة : وبعد ذلك أكل كتابة باقى الجملة .

وسوف يظهر السطر كاملاً كما يلي :

```
>20 PRU\U\INT "RADIUS=";
```

لاحظ أن الحرف المحذوف U مبين بين زوج من الشرط العكسية (⌘) . وتطبع الشرط العكسية أوتوماتيكياً عند بدء استخدام مفتاح DELETE وعند الانتهاء منه . ومع ذلك ، فن المهم أن نفهم أن الشرط العكسية والحروف المحذوفة لن ترسل إلى الحاسب . وعلى ذلك سوف يقوم الحاسب بتفسير السطر المكتوب كما هو مطلوب :

```
20 PRINT "RADIUS=";
```

مثال ٣ - ٥

دعنا نأخذ في الاعتبار مرة أخرى خطأ الطباعة الذي تمت مناقشته في المثال السابق . والآن نفترض أن المبرمج لم يلاحظ خطأه إلا بعد كتابة عدة حروف قليلة . وبذلك فإن السطر المكتوب يظهر :

```
>20 PRUNT "RA
```

عند اكتشاف الخطأ . فيجب على المبرمج أن يضغط على مفتاح DELETE 7 مرات وذلك لحذف كل شيء بدءاً من (ومنتصفنا) حرف U ، وبعد ذلك يمد كتابة باقى الجملة بطريقة صحيحة .

سوف يظهر السطر الأول مكتوباً كالتالى :

```
>20 PRUNT "RA\AR" TNU\INT "RADIUS=";
```

ومرة أخرى نرى أن الحروف المحذوفة محصورة بين زوج من الشرطات العكسية سوف تطيع الشرطتان أو توماتيكياً عند ابتداء استعمال مفتاح DELETE وعند الانتهاء منه . لاحظ أن الحذف يجرى بطريقة عكسية « أى أن أول حرف يحذف هو الحرف A ، ثم يتبعه الحرف R و . . . الخ . حتى يتم حذف الحرف U . وحيث أن هذه الحروف لم ترسل للحاسب فإن السطر سوف يخزن بهذه الطريقة :

```
20 PRINT "RADIUS=";
```

في بعض الأحيان لا يلحظ الخطأ إلا بعد كتابة السطر كاملاً (أو معظمه) في مثل هذا الموقف يمكن أن يكون حذف حرف في المرة الواحدة بمفتاح DELETE أو (مفتاح RUBOUT) عملية مرهقة جداً . ويكون الإجراء الأفضل هو حذف السطر كاملاً ، والاستعاضة عنه بسطر جديد . يمكن حذف السطر بالضغط على مفتاح ALTMODE أو مفتاح ESCAPE (المسمى ESC في شكل ٣ - ١) وذلك إذا لم يتم إرسال السطر للحاسب . أما إذا كان قد تم إرسال السطر فيمكن الاستعاضة عنه ببساطة بإدخال سطر جديد له نفس رقم الجملة كرقم السطر القديم .

مثال ٣ - ٦

عند إدخال البرنامج المبين في شكل ٣ - ٥ نفرض أن المبرمج قد كتب

```
>20 PRUNT "RA
```

وبعد ذلك اتضح أنه عمل خطأ (وذلك بطباعة U بدلا من I) . يمكن حذف السطر غير الصحيح بالضغط على مفتاح ESC وإعادة كتابة السطر صحيحاً بعد ذلك .

في طريقة أخرى يمكنه الضغط على مفتاح RETURN ، وبذلك يدخل الجملة غير صحيحة وغير كاملة للحاسب . ويمكنه بعد ذلك استكمال وكتابة الجملة الصحيحة أى :

```
>20 PRINT "RADIUS=";
```

عند الضغط على مفتاح RETURN مرة أخرى سوف يتم إدخال الجملة للحاسب وبذلك تستبدل الجملة السابقة (غير الصحيحة) التي لها نفس رقم الجملة .

وببساطة يمكن حذف جملة كاملة من برنامج ببسك بكتابة رقم الجملة وبعد ذلك بضغط مفتاح RETURN.

مثال ٣ - ٧

افرض أن البرنامج المبين في شكل ٣ - ٥ قد تم إدخاله للحاسب وبعد ذلك قرر المبرمج حذف جملة REM فيكتب :

```
>5
```

فقط ويضغط على مفتاح RETURN ، وبذلك تُلغى الجملة رقم 5 (جملة REM) .

٣ - ٥ تشغيل برنامج PROCESSING A PROGRAM

يكون البرنامج جاهزاً للتشغيل فور إدخاله للحاسب وتصحيح كل الأخطاء المروقة . وغالباً ما نريد إعادة كتابة البرنامج (أى عمل قائمة به) وتخزين البرنامج (أى الاحتفاظ به) لاستخدامه فيما بعد ، وبالطبع تنفيذ (أى تشغيل) البرنامج . تجرى هذه العمليات بسهولة بكتابة الكلمات LIST و SAVE و RUN . ويمكن للمبرمج إصدار هذه الأوامر بأي ترتيب يرغب فيه .

PITT DEC-1055/A 548.018 18:21:29

PLEASE LOGIN OR ATTACH.

.LOGIN

JOB 42 PITT DEC-1055/A 548.018 TTY42

#115421/160531

PASSWORD:

ALLOCATION REMAINING: 9.8 UNITS

1821 15-FEB THUR

.R BASIC

NEW OR OLD-->NEW

NEW FILE NAME-->CIRCLE

>20 PRINT\TNU\INT RADIUS

>30 INPUT R

>40 LET A=P*R^2

>50 LET C=2*P*R

>60 PRINT "R=";R,"A=";A,"C=";C

>70 GOTO 20

>80 END

>5 REM PROGRAM TO CALCULATE AREA AND CIRCUMFERENCE OF A CIRCLE

>10 LET P=3.1416\6\5927

>65 PRINT

>20 PRINT "RADIUS=";

>35 IF R=0 THEN 80

>LIST

CIRCLE 18:27 15-FEB

5 REM PROGRAM TO CALCULATE AREA AND CIRCUMFERENCE OF A CIRCLE

10 LET P=3.141593

20 PRINT "RADIUS=";

30 INPUT R

35 IF R=0 THEN 80

40 LET A=P*R^2

50 LET C=2*P*R

60 PRINT "R=";R,"A=";A,"C=";C

65 PRINT

70 GOTO 20

80 END

>SAVE

>RUN

CIRCLE 18:28 15-FEB

RADIUS= 715

R= 15 A= 706.858 C= 94.2478

RADIUS= 76.82

R= 6.82 A= 146.123 C= 42.8513

RADIUS= 737.4

R= 37.4 A= 4394.33 C= 234.991

RADIUS= 70

TIME: 0.21 SECS.

>BYE

JOB 42, USER[115421,160531] LOGGED OFF TTY42 1829 15-FEB

SAVED ALL FILES (25 BLOCKS)

CPUTIME 0:01 DISK R+W=77+15 CONNECT=8 MIN UNITS=0.0101

مثال ٣ - ٨ مساحة ومحيط دائرة

تم إدخال برنامج للحاسب مشابه للبرنامج المين في شكل ٣ - ٥ ويرغب المبرمج بعد ذلك في طباعة البرنامج وتخزينه وتنفيذه . وعلى ذلك يطبع المبرمج الكلمة LIST بعد الانتهاء من إدخال البرنامج الأصل (والذي يمكن أن يحتوى على تصحيح أخطاء ، وجمل غير مرتبة ، و . . . الخ) . ويترتب على ذلك إصدار قائمة للجمل بعد تصحيحها وفى الترتيب الصحيح .

فور انتهاء الحاسب من طباعة قائمة كاملة بالبرنامج ، يكتب المبرمج الكلمات SAVE و RUN من أجل تخزين وتشغيل البرنامج على الترتيب . وأخيراً يكتب المبرمج BYE بعد الانتهاء من التشغيل ، وبذلك ينهى اتصاله بالحاسب .

يبين شكل ٣ - ٦ قائمة بملزمة مشاركة زمنية كاملة . وكلما هو معتاد وضمت خطوط تحت المعلومات التى أدخلها المبرمج . لاحظ أن جمل البرنامج الأصل تحتوى على تصحيحات أخطاء فى الجمل 10 ، 20 وأن الجملة 20 الأولى قد استبدلت بجملة مصححة بعد ذلك . وأيضاً ، ترى أن الجمل قد كتبت بالتسلسل الذى يتم التنفيذ على أساسه .

بمجرد أن يكتب المبرمج LIST نرى عنوان البرنامج (وهو هنا CIRCLE) قد طبع وتبعته قائمة بجمل البرنامج بتسلسل صحيح . مدخلات ومخرجات البيانات اللازمة لتنفيذ البرنامج موضحة بعد أمر RUN . وأخيراً يضع سطور أخيرة تحتوى على معلومات إحصائية تعطى مع إجراء تسجيل الخروج من النظام وذلك نتيجة لأمر BYE . (وسوف يذكر ذلك بتفصيل أكثر فى قسم ٣ - ٦) .

لا يمكن للمبرمج الاستمرار فى الاتصال بالحاسب بعد تسجيل خروجه عن النظام (إلا ، بالطبع ، فى حالة تسجيل دخوله للنظام مرة ثانية) .

ومع ذلك ، فإن برنامج CIRCLE سوف يتم تخزينه للاستخدام المتعاقب وذلك نتيجة لأمر SAVE . (يحتمل أن يخزن البرنامج على قرص أو شريط مغنط وليس فى الذاكرة الأساسية للحاسب ، ولا يهم هذا المبرمج فى شئ) .

إذا أريد تشغيل برنامج سبق تخزينه ، وليس برنامج يدخل لأول مرة ، فيجب على المبرمج أن يكتب OLD بدلا من NEW بعد إجراء تسجيل دخوله للنظام . وسوف يجب الحاسب بالرسالة :

OLD FILE NAME-->

وبعد ذلك يجب على المبرمج أن يكتب اسم البرنامج المخزن . وابتداء من هذه النقطة يمكن تشغيل البرنامج بالطريقة التى سبق شرحها .

مثال ٣ - ٩

نفرض أننا نرغب فى تنفيذ برنامج CIRCLE الذى سبق مناقشته فى مثال ٣ - ٨ بعد أن تم تخزينه سابقاً . يبين شكل ٣ - ٧ قائمة بملزمة مشاركة زمنية كاملة متضمنة إجراءات التسجيل للدخول والخروج عن النظام . لاحظ أن الأوامر تشبه إلى حد بعيد الأوامر المبينة فى شكل ٣ - ٦ والفرق الوحيد هو استخدام كلمة OLD بدلا من NEW .

من أجل استبدال برنامج قديم بنسخة أحدث تم تنقيحها (مع الاحتفاظ بنفس الاسم السابق) فببساطة يكتب المبرمج REPLACE ويتمهما اسم البرنامج (لاحظ أن ذلك يتسبب فى تدمير النسخة القديمة من البرنامج نظراً لأن الحاسب سوف يكتب عليها النسخة الأحدث) .

تمكننا لغة البيسك أيضاً من تشغيل البرنامج بطرق أخرى . فثلاً يمكننا إلغاء برنامج سبق تخزينه بكتابة UNSAVE ، أو يمكننا الحصول على قائمة بأسماء البرامج المخزنة بكتابة CATALOG . إجمالاً ، فإن الأوامر المستخدمة لتشغيل برنامج تسمى أوامر النظام (أو أوامر التنقيح) . وبالرغم من أن أوامر التنقيح المتاحة سوف تختلف من نسخة بيسك لأخرى ، إلا أن كل نسخ اللغة تحتوي على الخصائص الأساسية الموصوفة أعلاه . ملخص لأوامر النظام الشائعة الاستخدام مبينة في الملحق C .

٣ - ٦ التسجيل للخروج من النظام LOGGING OUT

لقد رأينا أن الإجراء المطلوب لإنهاء الاتصال بالحاسب (أو إجراء تسجيل الخروج عن النظام) تبدأ بكتابة الكلمة GOODBYE (أو ببساطة BYE) . فور إصدار هذا الأمر فإن الحاسب سوف يجيب بكتابة ملخص لإحصائى الجلسة الحالية للمشاركة الزمنية . وتتضمن الإحصائيات التاريخ وعدد الملفات (أى البرامج ومجموعات البيانات) التى تم الاحتفاظ بها وفترة الاتصال (أى الزمن) ووقت الحاسب الفعول المستخدم ، وغالباً التكلفة . وإجراء تسجيل الخروج عن النظام موضحة فى الأمثلة ٣ - ٨ و ٣ - ٩ (أنظر الشكلين ٣ - ٦ و ٣ - ٧) بجانب المثال ٣ - ١٠ أدناه .

PITT DEC-1055/A 548.01B 20:13:09

PLEASE LOGIN OR ATTACH.

.LOGIN
JOB 62 PITT DEC-1055/A 548.01B TTY42
#115421/160531
PASSWORD:
ALLOCATION REMAINING: 9.8 UNITS
2013 15-FEB THUR

.R BASIC

NEW OR OLD-->OLD
OLD FILE NAME-->CIRCLE

>RUN

CIRCLE	20:14	15-FEB
RADIUS= 717.45		
R= 17.45	A= 956.623	C= 109.642
RADIUS= 712.7		
R= 12.7	A= 506.707	C= 79.7965
RADIUS= 732.6		
R= 32.6	A= 3338.76	C= 204.832
RADIUS= 70		
TIME: 0.20 SECS.		

>BYE
JOB 62, USER[115421,160531] LOGGED OFF TTY42 2013 15-FEB
SAVED ALL FILES (25 BLOCKS)
CPU TIME 0:01 DISK R+W=98+6 CONNECT=3 MIN UNITS=0.0078

شكل ٣ - ٧

مثال ٣ - ١٠

لقد أكل المبرمج تشغيل البرنامج الخاص به وقد كتب كلمة BYE . وأجاب الحاسب بكتابة الأسطر الثلاثة من المعلومات المبينة فى شكل ٣ - ٨ وبعد ذلك قطع الاتصال بالنهاية الطرفية فور الانتهاء من ذلك .

```
>BYE
JOB 27, USER [115421,160531] LOGGED OFF TTY42 2128 8-FEB
SAVED ALL FILES. (15 BLOCKS)
CPU TIME 0:07 DISK R+W=856+8 CONNECT=17 MIN UNITS=0.0240
```

شكل ٣ - ٨

ينص أول سطر من المخرجات أن المستفيد الذي يحمل رقم حساب 115421 و 160531 قد تم تسجيل خروجه عن النظام من النهاية الطرفية رقم 42 في الساعة 9:28 مساءً (أي الساعة 2128) في يوم 8 فبراير. وبين السطر الثاني أن عدد 15 «كتلة» من المعلومات قد تم الاحتفاظ بها (حيث أن «كتلة» تحتوي على 640 حرفاً في هذه النسخة من البيسك بالذات). ونرى في السطر الثالث أن المطلوب لتشغيل هذا البرنامج 0.07 ثانية من وقت الحاسب (حقيقة، وقت المشغل المركزي)، وتم قراءة المعلومات من على جهاز قرص منطاطي للتخزين 856 مرة وتم الكتابة على القرص 8 مرات وأن جلسة المشاركة الزمنية بأكملها استغرقت 16 دقيقة بتكلفة 0.0240 وحدة.

٧ - ٣ تحليل الأخطاء ERROR DIAGNOSTICS

غالباً تبقى أخطاء البرمجة بدون اكتشاف حتى تعمل محاولة لتنفيذ البرنامج. وعلى أي حال، بمجرد إصدار أمر RUN، سوف يصبح وجود أخطاء معينة واضحاً، حيث أن مثل هذه الأخطاء سوف تمنع تفسير البرنامج، أي، تحويله إلى برنامج بلغة الآلة. وبالتحديد بعض الأخطاء الشائعة من هذا النوع مرجعها متغير غير محدد أو رقم جملة غير محددة أو عدم توازن لاقواس الناحية اليمنى أو اليسرى أو الفشل في إنهاء البرنامج بجملة END أو... الخ. تسمى مثل هذه الأخطاء أخطاء لغوية (أو أخطاء في تكوين الجملة).

سوف تولد معظم نسخ بييسك رسالة تحليل عند اكتشاف خطأ لغوي. هذه الرسائل ليست دائماً كاملة وواضحة في معناها، ولكن برغم ذلك تساعد في التعرف على طبيعة ومكان الخطأ.

مثال ٣ - ١١

يبين شكل ٣ - ٩ برنامج بييسك مشابه للبرنامج الممثل في شكل ٣ - ٦، ما عدا أنه تم تقديم عدة أخطاء لغوية متممة. رسائل التحليل التي تولد عند إصدار أمر RUN مبينة بوضوح. لاحظ أنه قد تم اكتشاف الأخطاء الموجودة في السطور 40 و 80 وإغفال جملة END ومع ذلك، فالخطأ في السطر 50 (مشير للمتغير B وليس P) لم يتم اكتشافه.

```
EX3.11      22:19      18-FEB
5  REM PROGRAM TO CALCULATE AREA AND CIRCUMFERENCE OF A CIRCLE
10 LET P=3.1415927
20 PRINT "RADIUS=";
30 INPUT R
40 LET A=(P*R^2
50 LET C=2*B*R
60 PRINT "R=";R,"A=";A,"C=";C
70 PRINT
80 GO TO 15

>RUN

EX3.11      22:19      18-FEB
? ILLEGAL FORMULA IN LINE 40
? UNDEFINED LINE NUMBER 15 IN LINE 80
? NO END INSTRUCTION

TIME: 0.18 SECS.
```

شكل ٣ - ٩

غالباً ما تكون الأخطاء اللغوية وأخطاء الكتابة واضحة عند حدوثها . بينما الأخطاء المنطقية تكون أكثر خداعاً . وهنا ينقل البرنامج بدقة تعليقات المبرمج خالية من أخطاء الكتابة والأخطاء اللغوية ، لكن المبرمج أمد الحاسب بمجموعة من التعليقات الغير صحيحة منطقياً .

أحياناً يكون الخطأ المنطقي نتيجة حالة يمكن التعرف عليها بواسطة الحاسب . مثل هذا الموقف ربما ينتج من توليد كمية عددية كبيرة زائدة (يتعدى أكبر رقم مسموح بتخزينه في الحاسب) أو من محاولة لحساب الجذر التربيعي لرقم سالب ، . الخ . سوف تولد رسائل التحليل في مواقف من هذا النوع ، لتجعل من السهل التعرف على الأخطاء وتصحيحها . تسمى رسائل التحليل هذه أخطاء التنفيذ ، تميز بينها وبين أخطاء التفسير التي تم وصفها مسبقاً ..

مثال ٣ - ١٢

يبين شكل ٣ - ١٠ برنامج ببسك لحساب الجذور الحقيقية للمعادلة التربيعية .

$$ax^2 + bx + c = 0$$

باستخدام الصيغة التربيعية

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

يخلو البرنامج تماماً من الأخطاء اللغوية . بينما لا يستطيع البرنامج أن يتعامل مع القيم السالبة لـ $(b^2 - 4ac)$ (أنظر قاعدة ٣ من القسم ٢ - ٧) . وعلاوة على ذلك ، يحتمل حدوث مصاعب رقمية إذا كان المتغير a له قيمة عددية صغيرة جداً أو قيمة عددية كبيرة جداً (أنظر القاعدة ٥ من القسم ٢ - ١) .

بعد قائمة البرنامج فاننا نرى المخرجات التي تم توليدها للقيم $a = 1$ و $b = 2$ و $c = 3$ ثم للقيم $a = 10^{-30}$ و $b = 10^{10}$ و $c = 10^{36}$. نحصل في الحالة الأولى على قيمة سالبة للقيمة $(b^2 - 4ac)$ وهذا هو السبب في رسالتى التحليل الأولى [لاحظ أنه تم استكمال الحسابات باستخدام القيمة المطلقة للقيمة $(b^2 - 4ac)$] تتسبب المجموعة الثانية من البيانات في تحديد قيمة كبيرة هائلة للمتغير x_2 وبذلك يتسبب في رسالة الطغح . (لاحظ أن البيانات المعطاه بواسطة المستخدم موضوع تحتها خط) .

٣ - ٨ تحديد الأخطاء المنطقية LOGICAL DEBUGGING

لقد رأينا أن الأخطاء اللغوية وبعض الأنواع من الأخطاء المنطقية سوف تتسبب في توليد رسائل تحليلية أثناء ترجمة أو تنفيذ البرنامج . وتكون الأخطاء من هذا النوع سهلة في اكتشافها وتصحيحها . بينما تكون أخطاء المنطق عادة أكثر صعوبة في اكتشافها ، حيث المخرجات الناتجة في برنامج غير صحيح منطقياً يمكن أن تظهر خالية من الأخطاء . وعلاوة على ذلك ، فإن الأخطاء المنطقية غالباً ماتكون صعبة في اكتشافها حتى عندما يكون وجودها معلوماً (كثال ، عندما يكون من الواضح أن المخرجات المحسوبة غير صحيحة) . وعلى ذلك فيحتمل أن يلزم الأمر القيام بعمل « بوليسى » لاكتشاف وتصحيح الأخطاء من هذا النوع . مثل هذا العمل البوليسى يعرف بتحديد الأخطاء المنطقية .

```

EX3.12          22:05          22-FEB

10 REM REAL ROOTS OF A QUADRATIC EQUATION
20 INPUT A,B,C
30 IF A=0 THEN 90
40 LET D=B^2-4*A*C
50 LET X1=(-B+D^.5)/(2*A)
60 LET X2=(-B-D^.5)/(2*A)
70 PRINT "A=";A,"B=";B,"C=";C,"X1=";X1,"X2=";X2
80 GO TO 20
90 END

```

> RUN

```

EX3.12          22:05          22-FEB

```

? 1,2,3

* ABSOLUTE VALUE RAISED TO POWER IN LINE 50

* ABSOLUTE VALUE RAISED TO POWER IN LINE 60

```

A= 1          B= 2          C= 3          X1=0.414214 X2=-2.41421
? 1E-30,1E10,1E36

```

* OVERFLOW IN LINE 60

```

A= 1.00000E-30          B= 1.00000E+10          C=
1.00000E+36 X1=-8.32000E+32          X2=-1.70141E+38
? 0.0,0

```

TIME: 0.50 SECS.

شكل ٣ - ١٠

اكتشاف الأخطاء Detecting Errors

الخطوة الأولى لمعالجة الأخطاء المنطقية هو اكتشاف وجودها وأحياناً يمكن أن يتم إنجاز ذلك باعتبار البرنامج الجديد بيانات سبق معرفة إجاباتها. وإذا لم يتم الحصول على النتائج الصحيحة، فعلى ذلك أن البرنامج يحتوي على أخطاء. ومع ذلك، حتى إذا تم الحصول على النتائج الصحيحة فلا يمكن لنا أن نتأكد تمام التأكد من أن البرنامج خال من الأخطاء، حيث أن بعض الأخطاء تسبب نتائج غير صحيحة تحت ظروف معينة فقط (ومثال لذلك، قيم معينة من بيانات الإدخال أو مع أشياء اختيارية في البرنامج). ولذلك فيجب أن يجرى على البرنامج الجديد اختبارات دقيقة قبل التأكد أن هذا البرنامج تم تحديد أخطائه المنطقية. ويكون ذلك صحيحاً خصوصاً مع البرامج المعقدة أو البرامج التي سوف يتم استخدامها كثيراً بواسطة آخرين.

وكقاعدة عامة، يجب إجراء الحسابات يدوياً بمساعدة آلة حاسبة من أجل الحصول على نتائج معروفة. ومع ذلك ففي بعض المسائل فإن كمية العمل التي تتضمن إجراء الحسابات يدوياً تكون غير ممكنة. (فالمسألة التي تتطلب بضع ثواني من وقت حاسب كبير يمكن أن تتطلب عدة أسابيع لحلها يدوياً!) ولذلك فلا يمكن أحياناً تطوير عينة من المسألة لاختبار برنامج جديد. ورغم أن تحديد الأخطاء المنطقية لمثل هذه البرامج يمكن أن تكون صعبة فعلاً، فغالباً ما يستطيع المبرمج اكتشاف الأخطاء المنطقية بدراسة النتائج المحسوبة بدقة للتأكد من أنها معقولة.

تصحيح الأخطاء Correcting Errors

مجرد التأكد من أن البرنامج يحتوي على خطأ منطقي قد يتطلب الأمر شيئاً من الدهاء والبراعة لإيجاد الخطأ. ويجب دائماً أن يبدأ اكتشاف الخطأ بأن يراجع المبرمج كل مجموعة منطقية من الجمل في البرنامج بمنتهى الدقة. محصن بخلفية وجود خطأ في مكان ما بالبرنامج

وغالباً ما يستطيع المبرمج اكتشاف الخطأ بالدراسة الدقيقة . وإذا لم يكتشف الخطأ ربما يكون من الأصلح ترك البرنامج لفترة من الزمن . ليس أمراً غير عادي لمبرمج مُركز أكثر مما ينبغي إلا يرى خطأ واضحاً من أول مرة .

وإذا لم يتمكن المبرمج من تحديد الخطأ بعد تكرار فحص البرنامج ، فيجب عليه بعد ذلك أن يواصل ويميد تشغيل البرنامج . مع طباعة كمية كبيرة من المخرجات الوسيطة . ويشار إلى ذلك أحياناً بتتبع البرنامج وغالباً ما يتضح مصدر الخطأ فور فحص الحسابات الوسيطة بدقة .

وعندما ينتهي المبرمج من محاولة كل الحيل التي يمكن أن يفكر فيها ولم يجد الخطأ فربما يميل إلى التوهم بوجود خطأ بالماكينة أو خطأ في عملية الترجمة . بالرغم من ندرة ذلك فمن المحتمل حدوث مثل هذا الخطأ . (أحياناً تكون أخطاء الماكينة متقطعة . بينما أخطاء الترجمة تكون متصلة وبذلك تتكرر بنفس الصورة) . ومع ذلك ، ففي بعض الحالات يتم فور التوصل إلى الحل النهائي اكتشاف أن أخطاء الترجمة أو أخطاء الماكينة المشتبه فيها إنما ترجع إلى وجود خطأ منطوق .

وأخيراً يجب أن نعرف القارئ حقيقة أن الأخطاء المنطقية لا يمكن الهرب منها في برمجة الحاسب ، إلا أن المبرمج الواعي يجب أن يعمل كل محاولة للتقليل من حدوثها وبذلك فيجب أن يتوقع المبرمج أن كمية معينة من تحديد الأخطاء مطلوبة كجزء من الجهد الكلي في كتابة برنامج بيسك حقيقي وله معنى .

مثال ٣ - ١٣

مطلوب من طالب كتابة برنامج بيسك لحساب قيمة الصيغة :

$$y = \left(\frac{x-1}{x}\right) + \frac{1}{2}\left(\frac{x-1}{x}\right)^2 + \frac{1}{3}\left(\frac{x-1}{x}\right)^3 + \frac{1}{4}\left(\frac{x-1}{x}\right)^4 + \frac{1}{5}\left(\frac{x-1}{x}\right)^5$$

ولتبسيط عملية البرمجة فقد عرف الطالب متغيراً جديداً u :

$$u = \left(\frac{x-1}{x}\right)$$

وبذلك أصبحت الصيغة الرياضية كالتالي :

$$y = u + \frac{1}{2}u^2 + \frac{1}{3}u^3 + \frac{1}{4}u^4 + \frac{1}{5}u^5$$

شكل ٣ - ١١ يبين برنامج بيسك كامل لحساب هذه الصيغة :

```
10 PRINT "X=";
20 INPUT X
30 LET U=X-1/X
40 LET Y=U+(U/2)^2+(U/3)^3+(U/4)^4+(U/5)^5
50 PRINT "Y=";Y
60 END
```

>RUN

EX3.13

21:25

05-MAY

X= 72
Y= 2.20971

TIME: 0.08 SECS.

شكل ٣ - ١١

يعلم الطالب أن لا يجب أن تأخذ القيمة 0.69 عندما $x=2$ ، ومع ذلك ، في النتائج المحسوبة $y=2.20971$ عندما تكون $x=2$ ، كما هو مبين في شكل ٣ - ١١ . وينتهي الطالب إلى أن البرنامج يحتوي على أخطاء منطقية يجب اكتشافها وتصحيحها .

وبالفحص الدقيق للبرنامج يصبح الطالب مدركاً أن الجملة رقم 30 تنتج قيمة $u=1.5$ عندما تكون $x=2$ بينما القيمة الصحيحة يجب أن تكون $u=0.5$ والسبب في الخطأ هو حذف الأقواس في الجملة رقم 30 التي يجب أن تقرأ :

30 LET U = (X - 1)/X

وبعد ذلك صحح الطالب البرنامج وأعاد تشغيله للقيمة $x=2$. جملة PRINT (السطر رقم 50) قد تم تغييرها أيضاً وبذلك يمكن طبع القيمة المحسوبة للمتغير u بجانب القيمة المحسوبة للمتغير y .

```
10 PRINT "X=";
20 INPUT X
30 LET U=(X-1)/X
40 LET Y=U+(U/2)+2+(U/3)+3+(U/4)+4+(U/5)+5
50 PRINT "U=";U,"Y=";Y
60 END
>RUN
```

```
10 PRINT "X=";
20 INPUT X
30 LET U=(X-1)/X
40 LET Y=U+(U+2)/2+(U+3)/3+(U+4)/4+(U+5)/5
50 PRINT "U=";U,"Y=";Y
60 END
>RUN
```

EX3.13 21:27 05-MAY
X= 72
U= 0.5 Y= 0.567384

EX3.13 21:29 05-MAY
X= 72
U= 0.5 Y= 0.688542

TIME: 0.07 SECS.

TIME: 0.08 SECS.

شكل ٣ - ١٣

شكل ٣ - ١٢

عند تنفيذ البرنامج تم حساب قيمة u الصحيحة ، ولكن قيمة y مازالت غير صحيحة ، كما هو واضح في شكل ٣ - ١٢ . ولذلك . يستنتج الطالب أن البرنامج يحتوي على خطأ إضافي ، موجود في مكان ما في الجملة رقم 40 .

وبعد بعض الدراسات الإضافية اكتشف الطالب أن الجملة رقم 40 حقيقة غير صحيحة . ويجب أن تكتب هذه الجملة كما يلي :

40 LET Y=U+(U+2)/2+(U+3)/3+(U+4)/4+(U+5)/5

(لاحظ أن أزواج الأقواس غير ضرورية نتيجة للتدرج الطبيعي للعمليات) . وبعد ذلك تم تصحيح البرنامج كما هو مبين في شكل ٣ - ١٣ . ترى في نهاية شكل ٣ - ١٣ أن قيمة y المناظرة لقيمة $x=2$ قد تم حسابها بالصورة الصحيحة والنتيجة $y=0.688542$.

٣ - ٩ ملاحظات ختامية CLOSING REMARKS

نود أن نذكر القارئ مرة ثانية أن برمجة البيسك هي مهارة تشبه إلى حد كبير تعلم العزف على آلة موسيقية ، حيث لا يمكن تعلمها بسهولة بمجرد قراءة كتاب . ولذلك لابد للقارئ أن ينشغل بنشاط في كتابه برامجه الخاصة وتنقيح ، وتخزينها ووضعها في قائمة وتنفيذها وتحديد أخطائها . وفي نهاية هذا الفصل يجد القارئ قائمة بالعديد من البرامج المناسبة ، والمقترحة في مسائل البرمجة . ونهيب بالقارئ أن يقوم بحل العديد من هذه المسائل ما أمكنه ذلك .

بالرغم من أن مادة هذا الفصل تتجه نحو استخدام الحاسبات الضخمة والتي تعمل في أسلوب المشاركة الزمنية ، إلا أنها مفيدة لكل المبرمجين المبتدئين بصرف النظر عن البيئة الحاسوبية الخاصة بهم .

ومع كل فإننا ندعو مبرمجي الحاسبات الدقيقة بدراسة الأمثلة في قسماً ١٠ - ٢ ، ١٠ - ٧ بالإضافة إلى المادة الموجودة في هذا الفصل ، وذلك قبل محاولتهم تشغيل برامج البيسك الخاصة بهم .

اسئلة للمراجعة

Review Questions

- ٣ - ١ ما هي النهاية الطرفية المركزية (الكونسول أو نفذ التشغيل) . وما هي النهاية الطرفية للمشاركة الزمنية (مودم) ؟ وما هي النهاية الطرفية الذكية « البارة » ؟
- ٣ - ٢ صف ، بمصطلحات عامة ، إجراءات التسجيل المستخدمة مع البيسك للدخول إلى النظام والخروج منه .
- ٣ - ٣ ما هو المقصود بالتلقين ؟ وكيف يشار إلى التلقين على النهاية الطرفية عند إدخال أو تشغيل برنامج بيسك ؟
- ٣ - ٤ كيف يتم نقل سطر من المعلومات تم طباعتها على النهاية الطرفية للحاسب ؟
- ٣ - ٥ افترض أن برنامج بيسك تم إدخاله للحاسب ببعض جمل في أماكن غير صحيحة . فكيف يتم تصحيح مثل هذا الموقف ؟
- ٣ - ٦ كيف يمكن إلغاء حرف أو أكثر من سطر على النهاية الطرفية ؟ وكيف يمكن إلغاء سطر بالكامل ؟
- ٣ - ٧ كيف يمكن تغيير جملة غير صحيحة فور إرسالها للحاسب ؟
- ٣ - ٨ كيف يمكن إلغاء جملة من برنامج بيسك فور الانتهاء من إرسالها للحاسب ؟
- ٣ - ٩ كيف يمكن استصدار قائمة ببرنامج على النهاية الطرفية ؟
- ٣ - ١٠ كيف يمكن لبرنامج بيسك أن يخزن على أي وحدة مغناطيسية لاستخدامها فيما بعد ؟
- ٣ - ١١ كيف ينفذ برنامج البيسك ؟
- ٣ - ١٢ افترض أنه تم تخزين برنامج بيسك على وحدة تخزين مغناطيسية والمطلوب استرجاعها لتشغيلها مرة أخرى . كيف يمكن التوصل للبرنامج ؟
- ٣ - ١٣ صف الغرض من كل من أوامر النظام التالية :

BYE, CATALOG, GOODBYE, LIST, NEW, OLD, REPLACE, RUN, SAVE, UNSAVE

- ٣ - ١٤ ما هو المقصود بالأخطاء اللغوية ؟
- ٣ - ١٥ كيف تختلف الأخطاء اللغوية (النحوية) والأخطاء المنطقية كل عن الآخر ؟
- ٣ - ١٦ أذكر بعض الأخطاء اللغوية الشائعة .
- ٣ - ١٧ أذكر بعض الأخطاء المنطقية الشائعة .
- ٣ - ١٨ ما هو المقصود من رسائل التحليل وكيف يمكن التمييز بين تحليل أخطاء التفسير وتحليل أخطاء التنفيذ ؟
- ٣ - ١٩ هل تولد رسائل تحليل رداً على الأخطاء المنطقية ؟

٣ - ٢٠ ما هو المقصود من تحديد الأخطاء منطقياً ؟ أذكر بعض إجراءات تحديد الأخطاء المنطقية الشائعة .

٣ ٢١ بأى شكل يشبه تعلم كتابة برنامج ببسك تعلم الدق على الطبول ؟

مسائل تكميلية

Supplementary Problems

« المسائل » التالية تتعلق بتجميع المعلومات أكثر منها على حل المسائل حقيقة .

٣ - ٢٢ اجعل نفسك ملماً بالنهايات الطرفية للمشاركة الزمنية المستخدمة في مدرستك أو مكتبك .

(أ) أين مفتاح ON/OFF لتوصيل الكهرباء وقطعها ؟

(ب) هل بالوحدة جهاز تليفون بقرص مبنى داخلياً ؟ وإن كان كذلك ، كيف يعمل ؟

(ج) هل يوجد بالوحدة ذكاء مبنى داخلياً ؟

(د) ما هو المفتاح الذى يسبب إلغاء حرف أو أكثر من سطر ؟ إلغاء سطر بالكامل ؟

(هـ) ما هو المفتاح الذى يسبب إرسال سطر للحاسب ؟

(و) هل يمكن للوحدة أن تعمل بأسلوب LOCAL (أى كوحدة مستقلة إذا أصبحت غير متصلة بالحاسب) ؟

وإن كان كذلك ؟ كيف يمكن عمل ذلك ؟

٣ - ٢٣ حدد الإجراءات المستخدمة لتسجيل الدخول للنظام والخروج منه في مدرستك أو مكتبك . وما هى المعلومات المطلوبة للدخول للنظام تماماً ؟ ما معنى المعلومات الصادرة من الحاسب أثناء إجراءات التسجيل للدخول للنظام والخروج منه ؟

٣ - ٢٤ ما هى أوامر نظام ببسك المستخدم في مدرستك أو مكتبك ؟ اجعل نفسك ملماً بالأوامر التى لم تتم مناقشتها في هذا الفصل .

٣ - ٢٥ كيف يمكن التمييز بين تحليل أخطاء الترجمة وتحليل أخطاء التنفيذ في نسخة ببسك المستخدمة في مدرستك أو مكتبك ؟

٣ - ٢٦ كيف يتم تحديد تكلفة المشاركة الزمنية في مدرستك أو مكتبك ؟ وما هى التكلفة الحقيقية لوحدة واحدة من المشاركة الزمنية ؟

(لاحظ أنه من المهم أن يجاب على هذا السؤال بواسطة الطالب الذى يتلقى توصيل الحاسب بدون مقابل من خلال معهد تعليمي .

إن معظم الطلبة ليس لديهم أى فكرة عن التكلفة التجارية المساوية لاستخدام الحاسب !) .

مسائل للبرمجة

Programming Problems

٣ - ٢٧ اتصل بنظام الحاسب ، حدد أسماء أى برامج تم تخزينها تحت رقم الحاسب اخاص بك ثم انفصل عن النظام .

٣ - ٢٨ أدخل البرنامج المبين فى شكل ٣ - ٦ لحساب مساحة ومحيط الدائرة تأكد من تصحيح أى خطأ كتابى . اطلب قائمة بالبرامج بعد أن تم قراءته بواسطة الحاسب . وعند التأكد من أنه صحيح ، نفذ البرنامج عدة مرات باستخدام قيم تراها مناسبة لنصف القطر . تحقق من أن الإجابات المحسوبة صحيحة وذلك بمقارنتها بالنتائج المحسوبة يدوياً لاحظ أن تنهى التنفيذ عند إدخال قيمة نصف القطر صفر) .

٣ - ٢٩ أدخل ، وصحح ، واعمل قائمة ، ونفذ واحتفظ بالبرامج لقليل من المسائل التالية :

(أ) مسألة « أهلا » hello الموصوفة فى ٢ - ٦٢ (أ) .

(ب) مسألة « ما هو إسمك » ؟ what's your name ? الموصوفة فى ٢ - ٦٢ (ب) .

(ج) مسألة تحويل درجة الحرارة الموصوفة فى ٢ - ٦٣ (أ) .

(د) مسألة الحصالة الموصوفة فى ٢ - ٦٣ (ب) .

(هـ) حسابات حجم ومساحة الكرة الموصوفة فى ٢ - ٦٤ (أ) .

(و) حساب مساحة المثلث ، مساحات الدائرة الداخلية الكبرى والدائرة الخارجية الصغرى كما هو موصوف فى ٢ - ٦٤ (ج) .

(ز) مسألة الربيع المركب ، كما هى موصوفة فى ٢ - ٦٤ (د) .

(ح) حساب نمو البكتيريا المستزرعة كما هو موصوف فى ٢ - ٦٤ (هـ) .

الفصل ٤

التفرع وتكوين حلقات تكرارية Branching and Looping

كانت مسائل البرمجة التي درسناها حتى الآن من أنواع « آلة جمع » ومعنى هذا أن الحسابات كانت تتم دائماً بترتيب ثابت . ومع ذلك فإن الاستمالات المتعددة للحاسب الرقوى والجديرة بالملاحظة لا تقع تحت طائلة قدرته على القيام بعدة عمليات حسابية موصوفة في فترة قصيرة من الوقت ، ولكن في الواقع قدرته على عمل قرارات منطقية ثم تنفيذ مجموعة من الأوامر المناسبة ، معتمداً على نتائج هذه القرارات يجعل الحاسب مفيداً جداً . دعنا الآن نحول اهتمامنا إلى هذا الموضوع الهام .

لقد تعلمنا أن عمليات التفرع غير المشروط (أى ، تحويل التحكم ، أو « القفر » من جزء في البرنامج إلى آخر) يمكن أن تتم في البيسك بواسطة جملة GO TO (انظر قسم ٢ - ١٤) . هناك موقف آخر يبرز دائماً وهو تحويل التحكم إلى مكان من مكانين في البرنامج ، معتمداً على نتيجة مقارنة بين كيتين مثل هذه العملية تسمى عملية التفرع المشروط ، وتسمح باتخاذ قرارات منطقية داخل الحاسب .

هناك عملية أخرى تطلب دائماً في برنامج الحاسب ، وهى تكوين حلقات تكرارية . وهذا يتضمن تكرار بعض الأجزاء من البرنامج إما عدداً معيناً من المرات أو حتى استيفاء شرط معين . يمكن أن يحتوى الجزء المكرر من البرنامج (الحلقة التكرارية) على عملية تفرع مشروطة تقرر إنهاء الحلقة التكرارية أو تنفيذها على الأقل مرة واحدة أخرى . أما إذا تطلب الأمر إعادة تنفيذ الحلقة التكرارية مرة أخرى ، فيتحوّل التحكم لبداية الحلقة التكرارية . وبذلك فلا داعى لكتابة الأوامر في الجزء المكرر من البرنامج أكثر من مرة .

سوف نرى في هذا الفصل كيف يتم القيام بعمليات التفرع والتكرار في البيسك وهذا يفتح الباب إلى طراز أوسع وأكثر تشويقاً . من المسائل المبرمجة .

٤ - ١ المعاملات الرابطة RELATIONAL OPERATORS

يجب أن نجد طريقة للتعبير عن شروط التساوى أو عدم التساوى من أجل القيام بعملية التفرع المشروط في البيسك ويتم إنجازها من خلال استعمال المعاملات الرابطة . وهذه المعاملات هى :

$A = B$	A تساوى B
$A <> B$	A لا تساوى B
$A < B$	A أقل من B
$A < = B$	A أقل من أو تساوى B
$B > A$	B أكبر من A
$B > = A$	B أكبر من أو تساوى A

تستخدم المعاملات الرابطة لتصل بين كيات عديدة (أى ، أرقام ، متغيرات أو صيغ رياضية) أو سلاسل من الحروف ، وبذلك تكون الشروط إما مستوفاة أو غير مستوفاة .

مثال ٤ - ١

فيما يلي نين عدة شروط بها كيات عددية . كل منها ، إما أن يكون مستوفياً أو غير مستوف ، معتمداً في ذلك على القيمة العددية للمتغيرات .

```
X=27
N<=,001
C>(C1+C2)^2
A+B<C+D
P<>Q
Z>=X*Y
```

وبذلك يكون الشرط الأخير مستوفياً إذا كانت قيمة Z أكبر من أو تساوى قيمة حاصل ضرب X*Y فيما عدا ذلك يكون الشرط غير مستوف .

تفسر الشروط غير المتساوية التي تحتوى على سلاسل حرفية بطريقة « القادماً قبل » أو « القادماً بعد » وليس « أقل من » أو « أكبر من » علاوة على ذلك ، فإن الفراغات التي تلى السلسلة الحرفية يتم تجاهلها عند مقارنة السلاسل الحرفية .

مثال ٤ - ٢

نعرض فيما يلي عدة شروط تحتوى سلاسل حرفية . كل شرط إما أن يكون مستوفياً أو غير مستوف ، معتمداً في ذلك على القيمة الحرفية المعطاة للمتغيرات الشكلية .

```
N$="SMITH"
P$<>Q$
C$<G$
```

سوف يكون الشرط الأول مستوفياً إذا كانت القيمة المعطاة للمتغير N\$ هي SMITH . وإلا سيكون الشرط غير مستوف . وسيكون الشرط الثانى مستوفياً إذا كانت القيمة المعطاة للمتغير P\$ مختلفة عن القيمة المعطاة للمتغير Q\$ من أجل استيفاء الشرط الأخير يجب أن تأتى القيمة المعطاة للمتغير C\$ قبل القيمة المعطاة للمتغير G\$ في قائمة الحروف الأبجدية .

٤ - ٢ التفرع المشروط — جملة IF-THEN

CONDITIONAL BRANCHING—THE IF-THEN STATEMENT

تستخدم جملة IF-THEN للقيام بعمليات التفرع المشروط . تتكون الجملة من الكلمتين IF و THEN مفصولة بعلاقة ما ، ويتبعها رقم لجملة بعيدة . وعند تنفيذ جملة IF-THEN سوف يتحول التحكم إلى الجملة البعيدة إذا استوفت العلاقة ، وإلا ، فسوف تنفذ الجملة التي تلى جملة IF-THEN . (لاحظ أنه يمكن تحويل التحكم إلى أى جملة بعيدة بداخل البرنامج ، حتى إذا كانت جملة REM) .

مثال ٤ - ٣

مبين فيما يلي جزء من برنامج يبسك يحتوى على جملة IF-THEN

```
15...
...
50 IF I>=100 THEN 80
55 LET I=I+1
60 GO TO 15
...
80...
```

الطريقة التي سوف يتم بها تنفيذ البرنامج متوقفة على العلاقة :

$$I > 100$$

الموجودة في جملة IF-THEN إذا استوفت العلاقة الشرط (أى إذا كانت قيمة I أكبر من أو تساوى 100) ، فإن الجملة رقم 80 سوف تنفذ بعد ذلك ، وإذا كانت العلاقة غير مستوفاة (أى أن قيمة I أقل من 100) فإن الجملة التي سوف تنفذ بعد ذلك هي الجملة رقم 55 .

لاحظ الطريقة التي تستخدم فيها جملة IF-THEN مقترنة بجملة GO TO في هذا المثال لتكوين حلقة تكرارية .

كثير من نسخ بيسك تسمح باستخدام الكلمات GO TO بدلا من استخدام THEN .

مثال ٤ - ٤

فيما يلي جملة IF-GO TO متضمنة سلاسل حرفية :

45 IF N\$="SHARON" GO TO 120

(يجب أن يكون واضحا أن استخدام GO TO بدلا من THEN ليس له أى علاقة بنوع العلاقة ، أى إذا كانت كميات عددية أو سلاسل حرفية) .

مثال ٤ - ٥ : جذور معادلة جبرية Roots of an Algebraic Equation

غالبا ما تستخدم الحاسبات لحل المعادلات الجبرية التي لا يمكن حلها بطرق أولية . فلندرس مثلا ، هذه المعادلة :

$$x^5 + 3x^2 - 10 = 0$$

لا يمكن إعادة ترتيب المعادلة فتنج حلا مهيحاً للمتغير x ولذلك فسوف نحدد الحل بواسطة تكرار إجراء المحاولة والخطأ (أى ، إجراء تكرارى) والتي ينتج عنها تنقيح متوال للحل الذي تم التوصل إليه بالتخمين أولا .

إجراء حسابي Computational Procedure

سوف نبدأ بإعادة ترتيب المعادلة في الصيغة :

$$x = \sqrt[5]{10 - 3x^2}$$

وإجراؤنا سيكون بتخمين قيمة للمتغير x ، ونعوض بالقيمة في الطرف الأيمن من المعادلة التي تم إعادة ترتيبها وبذلك نحسب قيمة جديدة للمتغير x . وهذه القيمة الجديدة سنعوض بها في الطرف الأيمن بعد ذلك ، ونحصل على قيمة أخرى للمتغير x ، وهكذا . وسوف نستمر في الإجراء إما حتى تصبح القيم المتتالية للمتغير x قريبة قريبا كافيا (أى أن الطريقة تقاربت) أو حتى يتم عمل عدد معين من التكرارات (بذلك نمنع استمرار الحسابات بصورة لا تنتهى في حالة ما إذا كانت النتائج المحسوبة لا تتقارب)

ولتر كيف تعمل الطريقة ، لنفرض أننا اخترنا قيمة أولية للمتغير x وهي 1.0 . وبالتعويض بهذه القيمة في الطرف الأيمن من المعادلة ، نحصل على :

$$x = \sqrt[5]{10 - 3(1.0)^2} = 1.47577$$

ونعوض بهذه القيمة الجديدة للمتغير x في المعادلة سوف نحصل على :

$$x = \sqrt[5]{10 - 3(1.47577)^2} = 1.28225$$

وبتكرار هذا الإجراء نحصل على :

$$x = \sqrt[5]{10 - 3(1.28225)^2} = 1.38344$$

$$x = \sqrt[5]{10 - 3(1.38344)^2} = 1.38613$$

وهكذا . لاحظ أن القيم المتتالية للمتغير x تتقارب لقيمة نهائية .

مخطط تمهيدى لبرنامج The Program Outline

من أجل كتابة مخطط تمهيدى لبرنامج ببسك دعنا نعرف الرموز التالية :

X = قيمة المتغير x الذى نعوض به فى الطرف الأيمن من المعادلة .

$X1$ = القيمة الجديدة المحسوبة للمتغير x .

I = عدد لحساب مرات التكرار (سوف تزداد قيمة I بمقدار واحد فى كل مرة تكرار تالية) .

N = الحد الأقصى المسموح به لعدد مرات التكرار .

وسوف نستمر فى الحسابات حتى يتم الآتى :

(١) إما أن يصبح الفرق بين قيمتين متتاليتين للمتغير x أقل من 0.00001 ، أو ،

(ب) أن يصل عدد التكرار 1 للحد الأقصى المسموح به أى للقيمة (N)

والآن يمكننا أن نكتب مخططاً تمهيدياً لبرنامج ببسك كما يلى :

١ - اقرأ قيمتى X و N

٢ - ضع قيمة أولية فى العداد I (ضع به القيمة 1) .

٣ - احسب قيمة المتغير $X1$ باستخدام الصيغة الرياضية :

$$X1 = (10 - 3 * X + 2)^{.2}$$

٤ - اطبع القيم الجديدة المحسوبة للمتغيرين $X1$ و I (وبطباعة النتائج لكل تكرار بهذه الطريقة فإننا نرى حقيقة ما إذا كانت الحسابات تتقارب أم لا) .

٥ - اختبر قيمة $|X - X1|$ (أى القيمة المطلقة للقيمتين المتتاليتين للمتغير x لتحديد ما إذا كانت أقل من 0.00001 أم لا) :

(١) إذا كانت $|X - X1|$ أقل من 0.00001 ، فاذهب للخطوة 7 (اطبع النتائج النهائية) .

(ب) إذا كانت $|X - X1|$ أكبر من أو تساوى 0.00001 ، فتقدم للخطوة 6 التالية .

٦ - اختبر قيمة I إذا كانت تساوى N (لاحظ أن قيمة I سوف تكون أقل من N فى المراحل الأولى من الحسابات) .

(١) إذا كانت I تساوى N فاذهب للخطوة 8 (اطبع رسالة تشير إلى أن الحسابات لم تتقارب) .

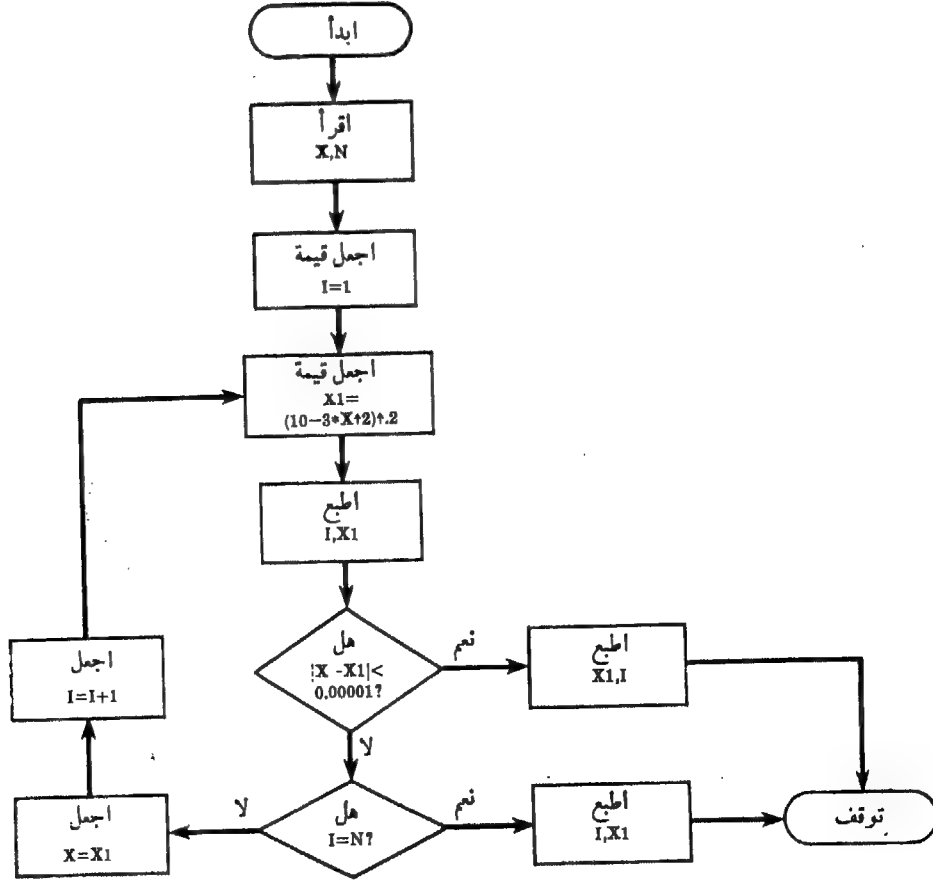
(ب) إذا كانت I أقل من N ، فزد قيمة I بمقدار واحد أى ($I = I + 1$) اجعل القيمة التى حسبت أخيراً للمتغير $X1$ تسمى X واذهب مرة أخرى للخطوة 3 ، وبذلك تبدأ التكرار التالى .

٧ - اطبع القيم النهائية للمتغيرين $X1$ و I ، وبعد ذلك اذهب للخطوة ٩ (توقف) .

٨ - اطبع رسالة تشير إلى أن الحسابات لم تتقارب واتبها بأحدث القيم للمتغيرين $X1$ و I .

٩ - توقف .

يبين شكل ٤ - ١ خريطة لسير عمليات إجراء الحسابات .



شكل ٤ - ١

برنامج البيسك The BASIC Program

نرى في شكل ٤ - ٢ برنامج بيسك كاملاً يناظر المخطط التمهيدى وخريطة سير العمليات السابقة . يحتوى البرنامج على جملة IF في السطور 60 و 70 . (وبفرض التوضيح فقد تم استخدام كل من الصيغ IF—THEN و IF—GO TO) . لاحظ أن البرنامج يقرأ قيمة مبدئية للمتغير X وقيمة للمتغير N ، ولكن معيار التقارب (أى القيمة 0.00001) قيمة ثابتة بداخل البرنامج ، وكان يمكن أن نعامل معيار التقارب كقيمة مدخلة إذا رغبتنا في ذلك . ولاحظ أيضاً ، استخدام الحروف ABS في السطر 60 . وذلك يشير إلى دالة بيسك المكتوبة التي تحدد القيمة المطلقة للكمية (X - X1) . (سوف نناقش الدوال المكتوبة في الفصل الخامس ، قسم ٥ - ١) . وأخيراً - نرى أنه كان من الممكن أن نواجه بعض الصعوبات بالبرنامج إذا تم حساب الصيغة الرياضية :

$$10 - 3 * X + 2$$

في السطر 40 وكانت النتيجة سالبة . ومن الممكن أن يتضمن البرنامج اختباراً لمثل هذا الشرط . لاحظ أن المعلومات المدخلة موضوع تحتها خط

>LIST

```

EX4.5          17:51          03-MAR
10 REM AN ITERATIVE METHOD FOR COMPUTING ROOTS OF AN EQUATION
20 INPUT X,N
25 PRINT
30 LET I=1
40 LET X1=(10-3*X^2)^.2
50 PRINT "I=";I,"X1=";X1
60 IF ABS(X-X1)<.00001 THEN 110
70 IF I=N GOTO 160
80 LET X=X1
90 LET I=I+1
100 GOTO 40
110 PRINT
120 PRINT "THE FINAL ANSWER IS X =";X1
130 PRINT
140 PRINT "NUMBER OF ITERATIONS REQUIRED =";I
150 GOTO 190
160 PRINT
165 PRINT "COMPUTATION HAS NOT CONVERGED AFTER ";I;" ITERATIONS"
170 PRINT
180 PRINT "LAST VALUE OF X =";X1
190 END

```

>RUN

```

EX4.5          17:52          03-MAR

```

71, 25

I= 1	X1= 1.47577
I= 2	X1= 1.28225
I= 3	X1= 1.38344
I= 4	X1= 1.33613
I= 5	X1= 1.35951
I= 6	X1= 1.34826
I= 7	X1= 1.35375
I= 8	X1= 1.35109
I= 9	X1= 1.35238
I= 10	X1= 1.35175
I= 11	X1= 1.35206
I= 12	X1= 1.35191
I= 13	X1= 1.35198
I= 14	X1= 1.35195
I= 15	X1= 1.35196
I= 16	X1= 1.35195

THE FINAL ANSWER IS X = 1.35195

NUMBER OF ITERATIONS REQUIRED = 16

TIME: 0.36 SECS.

شكل ٤ - ٢

يل قائمة البرنامج المخرجات التي تم توليدها لقيمة التخمين الابتدائية $x = 1.0$ لاحظ أن الحسابات قد تم تقاربها لحل $x = 1.35195$ بعد 16 تكرار. ومن النتائج المطبوعة يمكننا فعلاً رؤية أن قيم x المتتالية تتقارب وتتقارب حتى تصل لحل النهائي المتقارب.

يبين شكل ٤ - ٢ النتائج التي تم توليدها إن لم نحصل على التقارب. في هذه الحالة قد حددنا عدداً أقصى للتكرارات وهي 10 تكرارات فقط ($N = 10$). وهذا المدد غير كاف للحصول على حل تقاربي بالقيمة الابتدائية $x = 1.0$. يشار بوضوح إلى عدم الوصول إلى التقارب المطلوب بواسطة رسالة مطبوعة

>RUN

EX4.5

17:53

03-MAR

71, 10

I= 1	X1= 1.47577
I= 2	X1= 1.28225
I= 3	X1= 1.38344
I= 4	X1= 1.33613
I= 5	X1= 1.35951
I= 6	X1= 1.34826
I= 7	X1= 1.35375
I= 8	X1= 1.35109
I= 9	X1= 1.35238
I= 10	X1= 1.35175

COMPUTATION HAS NOT CONVERGED AFTER 10 ITERATIONS

LAST VALUE OF X = 1.35175

TIME: 0.26 SECS.

شكل ٤ - ٣

٤ - ٣ التفرع المتعدد — جملة ON — GO TO

MULTIPLE BRANCHING—THE ON-GO TO STATEMENT

يمكن القيام بالتفرع المتعدد في البيسك بواسطة جملة *ON-GOTO* تحتوى هذه الجملة على متغير رقمى أو صيغة رياضية ورقمين أو أكثر لحمل بميدة . سيتحول التحكم للجملة البعيدة الأولى إذا كانت قيمة المتغير أو الصيغة الرياضية تساوى 1 وإلى الجملة البعيدة الثانية إذا كانت قيمة المتغير أو الصيغة الرياضية تساوى 2... إلخ .

مثال ٤ - ٦

مبين فيما يلى جملة ON - GO TO نموذجية :

30 ON K GO TO 15,40,25,40,60

سوف يتحول التحكم إلى الجملة رقم 15 إذا كانت قيمة K = 1 وإلى الجملة رقم 40 إذا كانت قيمة K = 2 أو K = 4 وإلى الجملة رقم 25 إذا كانت قيمة K = 3 وإلى الجملة رقم 60 إذا كانت قيمة K = 5

إذا كانت قيمة المتغير أو الصيغة الرياضية ليست قيمة صحيحة فسوف يتجاهل الجزء العشرى من الرقم (أى أن الرقم سوف يبتز) .

مثال ٤ - ٧

نفرض أن القيمة التى أعطيت للمتغير K فى المثال ٤ - ٦ هى 3.67 . سوف يتجاهل الجزء 67 . وسوف نعتبر أن قيمة K هى 3 . وبذلك فسوف يتحول التحكم للجملة رقم 25 .

لاحظ أن القيمة الأصلية التى تم إعطاؤها للمتغير K قد تم بثرتها وليس تقريبها .

تسمح معظم نسخ البيسك باستخدام كلمة THEN بدلا من GO TO

مثال ٤ - ٨

نبين فيما يلي جملة ON — GO TO نموذجية :

50 ON A+2+B+2 GO TO 20,150,180

في معظم نسخ البيسك يمكن أن تكتب الجملة السابقة كما يلي :

50 ON A+2+B+2 THEN 20,150,180

{ — } جملة STOP THE STOP STATEMENT

تستخدم جملة STOP لإنهاء الحسابات عند أى نقطة من البرنامج . وهى مساوية لجملة GO TO التى تحول التحكم لجملة END. وتتكون الجملة ببساطة من رقم سطر وتبعتها الكلمة STOP

من المهم أن نفهم الاختلاف بين جملي STOP و END ، يمكن أن تظهر جملة STOP فى أى مكان من برنامج البيسك ، ما عدا فى آخر البرنامج . ويمكن أن تظهر أكثر من جملة STOP وعلى العكس فإن جملة END لا يمكن أن تظهر فى أى مكان ما عدا فى نهاية البرنامج ، وبذلك فلا يمكن استخدامها أكثر من مرة واحدة . فى البرنامج . (تذكر أن كل برنامج بيسك يجب أن ينتهى بجملة END) . مثال ٤ - ٩ التالى يوضح استخدام جملة STOP بجانب جملة ON — GO TO .

مثال ٤ - ٩ حساب قيمة الاستهلاك Calculating Depreciation

دعنا ندرس كيفية حساب قيمة الاستهلاك السنوى لعنصر تتناقص قيمته (مثل ، مبنى ، قطع آلات ، .. إلخ) . توجد ثلاث طرق مختلفة لحساب قيمة الاستهلاك تعرف بطريقة الخط المستقيم وطريقة توازن الانحراف المزدوج وطريقة مجموع أرقام السنوات . نود كتابة برنامج بيسك يسمح لنا بأختيار أى من هذه الطرق لكل مجموعة من الحسابات .

طريقة إجراء الحسابات Computational Procedure

سوف تبدأ الحسابات بقراءة القيمة الأصلية (بدون استهلاك) للعنصر ، مدة حياة العنصر (عدد السنوات التى على مداها يستهلك العنصر) ورقم صحيح يشير إلى طريقة الاستهلاك المستخدمة . وسوف تحسب وتطبع قيمة الاستهلاك السنوى .

طريقة الخط المستقيم وهى أسهل طريقة يمكن استخدامها . وفى هذه الطريقة ، تقسم القيمة الأصلية على مدة حياته (العدد الإيجابى للسنوات) . وغارج القسمة سوف يكون الكمية التى يستهلك بها العنصر كل سنة . فمثلا ، إذا كان عنصر ثمنه \$8000 سوف يستهلك على مدى عشر سنوات ، فإن الاستهلاك السنوى سوف يكون \$800 = \$8000 ÷ 10 . سوف تتناقص قيمة العنصر بمبلغ \$800 كل سنة . لاحظ أن الاستهلاك السنوى هو نفس الاستهلاك كل سنة

عند استخدام طريقة توازن الانحراف المزدوج ، فإن قيمة العنصر تتناقص بنسبة مئوية ثابتة كل سنة (وعلى ذلك فإن الكمية الحقيقية للاستهلاك بالدولار سوف تتغير من سنة لأخرى) . وللحصول على معامل الاستهلاك سوف تقسم ٢ على عمر العنصر . وبضرب هذا المعامل فى قيمة العنصر عند بداية كل سنة (ليست قيمة العنصر الأصلية) وذلك للحصول على الاستهلاك السنوى .

ولنفرض مثلا ، أننا نرغب فى تخفيض سعر عنصر ثمنه \$8000 على مدى عشر سنوات ، باستخدام طريقة توازن الانحراف المزدوج . سوف يكون معامل الاستهلاك $0.20 = 2 \div 10$ حيث تكون قيمة الاستهلاك للسنة الأولى عبارة عن $0.20 \times \$8000$ أى تساوى \$1600 وقيمة الاستهلاك للسنة الثانية تكون $(\$8000 - \$1600) \times 0.20$ أى تساوى 1280 $= 0.20 \times \$6400$ وقيمة الاستهلاك للسنة الثالثة تكون $1024 = 0.20 \times \$5210$ وهكذا .

فى طريقة مجموع أرقام السنوات سوف تتناقص قيمة العنصر بنسبة مئوية مختلفة فى كل سنة سوف يكون معامل الاستهلاك كسراً مقامه مجموع الأرقام من 1 إلى N حيث تمثل N طول حياة العنصر (مثلا ، حياة مدتها عشر سنوات سوف يكون المقام

$1 + 2 + 3 + \dots + 9 + 10 = 55$. ولأول سنة سيكون البسط N ، وللسنة الثانية سيكون البسط $(N - 1)$ وللسنة الثالثة سيكون البسط $(N - 2)$... وهكذا . ويحصل على الاستهلاك السنوي بضرب معامل الاستهلاك في القيمة الأصلية للعنصر .
ولتر كيف تعمل طريقة مجموع أرقام السنوات ، سوف يستهلك عنصراً قيمته \$ 8000 على مدى عشر سنوات . وسوف يكون الاستهلاك للسنة الأولى \$ 1454.55 $= \$ 8000 \times (10-55)$. ويكون الاستهلاك للسنة الثانية \$ 1309.09 $= \$ 8000 \times (9/55)$. وهكذا .

مخطط تمهيدى للبرنامج The Program Outline

نبدأ بتعريف الرموز التالية :

$$V = \text{قيمة العنصر .}$$

$$N = \text{عدد السنوات التي يستهلك فيها العنصر (حياته) .}$$

$$I = \text{رقم صحيح يشير للطريقة التي تستخدم لحساب الاستهلاك :}$$

$$(1) I = 1 \text{ تشير إلى حساب الاستهلاك بطريقة لخط المستقيم}$$

$$(ب) I = 2 \text{ تشير لطريقة توازن الانحراف المزدوج .}$$

$$(ج) I = 3 \text{ تشير لطريقة مجموع أرقام السنوات .}$$

$$J = \text{عداد يشير إلى السنة التي تدرس حالياً .}$$

$$D1 \text{ و } D2 \text{ و } D3 = \text{الاستهلاك السنوي المحسوب لكل من الطرق الثلاث .}$$

وسوف يتبع برنامج البيسك الخدس بنا المخطط التمهيدى الممثل أدناه .

$$1 - \text{اقرأ } V \text{ و } N \text{ و } I$$

$$2 - \text{اطبع رسالة تشير إلى الطريقة المستخدمة لحساب الاستهلاك}$$

$$3 - \text{ضع صفراً في العدد } J$$

$$4 - \text{احسب } D1 = V/N$$

$$F1 = \frac{V}{N*(N+1)/2}$$

(تستخدم $F1$ في طريقة مجموع أرقام السنوات . لاحظ أن مجموع الأرقام $1 + 2 + 3 + \dots + N$ مساوياً $(N*(N+1)/2)$.)

$$5 - \text{أضف 1 للعداد } J \text{ (أي } J = J + 1)$$

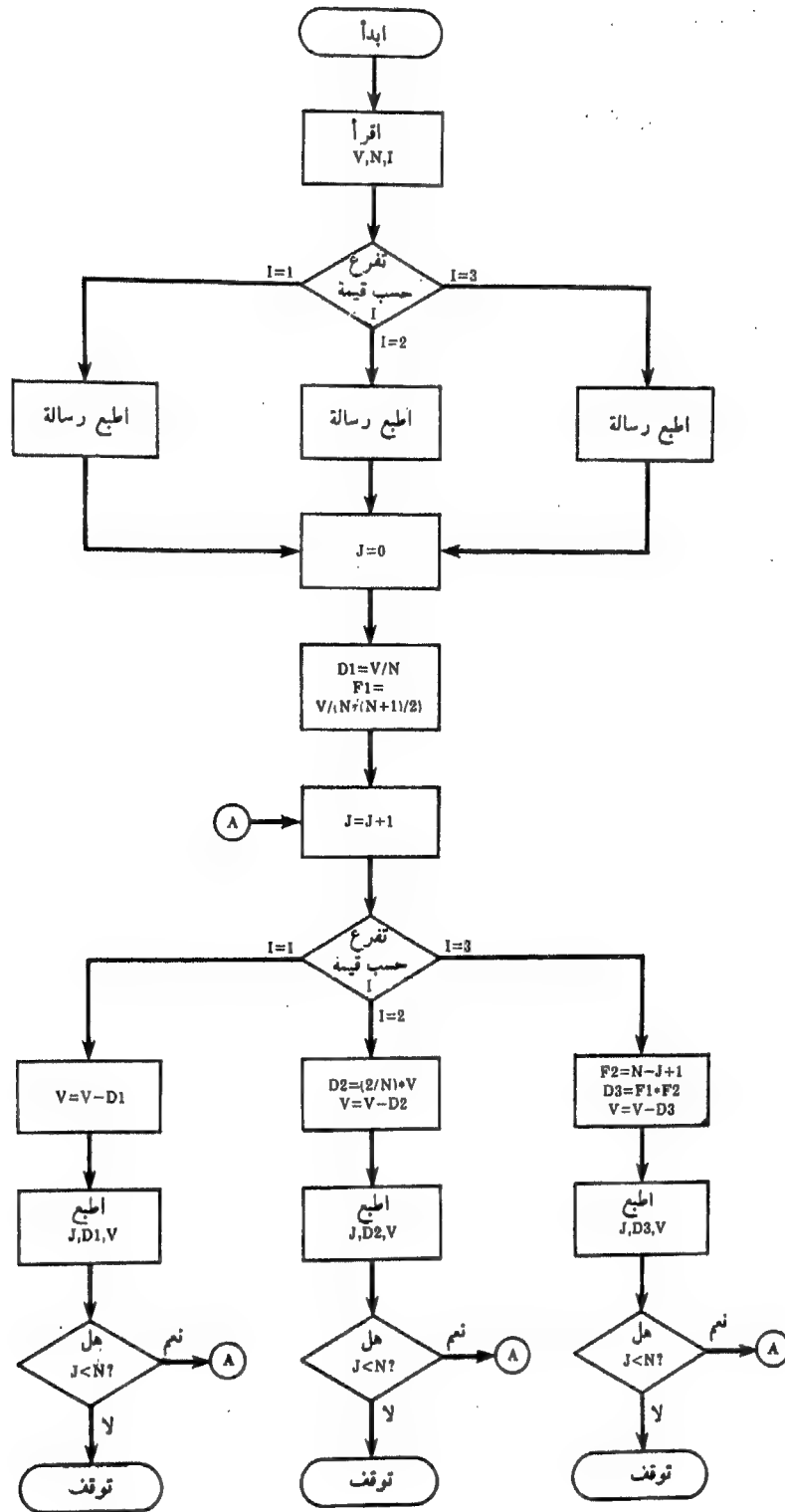
$$6 - \text{احسب الاستهلاك السنوي والقيمة الجديدة للعنصر بالطريقة المناسبة واطبع النتائج .}$$

$$(1) \text{ إذا كانت } I = 1 \text{ فننفذ ما يلي}$$

$$(i) \text{ احسب } V = V - D1$$

$$(ii) \text{ اطلع } J, D1, V$$

$$(iii) \text{ إذا كانت } J < N \text{ إذهب للخطوة 6 وإلا اوقف البرنامج .}$$



شكل ٤ - ٤

```

10 REM COMPUTATION OF DEPRECIATION BY THREE DIFFERENT METHODS
20 PRINT "V=";
30 INPUT V
40 PRINT "N=";
50 INPUT N
60 PRINT "I=";
70 INPUT I
80 PRINT
90 ON I GOTO 100,120,140
100 PRINT "STRAIGHT-LINE METHOD"
110 GOTO 150
120 PRINT "DOUBLE DECLINING BALANCE METHOD"
130 GOTO 150
140 PRINT "SUM-OF-THE-YEARS'-DIGITS METHOD"
150 PRINT
160 PRINT "END OF YEAR","DEPRECIATION","CURRENT VALUE"
170 LET J=0
180 LET D1=V/N
190 LET F1=V/(N*(N+1)/2)
200 LET J=J+1
210 ON I GOTO 300,400,500
215
300 REM STRAIGHT-LINE METHOD
305
310 LET V=V-D1
320 PRINT J,D1,V
330 IF J<N THEN 200
340 STOP
345
400 REM DOUBLE DECLINING BALANCE METHOD
405
410 LET D2=(2/N)*V
420 LET V=V-D2
430 PRINT J,D2,V
440 IF J<N THEN 200
450 STOP
455
500 REM SUM-OF-THE-YEARS'-DIGITS METHOD
505
510 LET F2=N-J+1
520 LET D3=F1*F2
530 LET V=V-D3
540 PRINT J,D3,V
550 IF J<N THEN 200
560 END

```

شكل ٤ - ٥

(ب) إذا كانت $I = 2$ فننفذ ما يلي :

(i) احسب $D2 = (2/N) * V$

$V = V - D2$

(ii) اطبع $J, D2, V$

(iii) إذا كانت $J < N$ اذهب للخطوة 5 ، وإلا أوقف البرنامج .

(ج) إذا كانت $I = 3$ فننفذ ما يلي :

$$F2 = N - J + 1 \text{ احسب (i)}$$

$$D3 = F1 * F2$$

$$V = V - D3$$

(ii) اطبع V و $D3$ و J

(iii) إذا كانت $J < N$ اذهب للخطوة 5 ، وإلا أوقف البرنامج .

يبين الشكل ٤ - ٤ خريطة سير العمليات المناظرة لذلك .

The BASIC Program برنامج بيسك

نرى في شكل ٤ - ٥ برنامج بيسك كامل للقيام بعمل الحسابات . لاحظ أن البرنامج يحتوى على جملة ON-GO TO في السطر 90 والسطر 210 (وكان يمكننا استعمال ON-THEN بنفس السهولة ، وسيكون المنطق متطابقاً) . تمدنا كل من هذه الجمل بشرط له ثلاثة تفرعات في هذه المسألة . وسنرى أيضاً فائدة جملة STOP في السطر 340 والسطر 450 وكان يمكننا استخدام GO TO كبدل وكتابة GO TO 550 في مكان جملة STOP

$$\begin{aligned} V &= 78000 \\ N &= 710 \\ I &= 72 \end{aligned}$$

DOUBLE DECLINING BALANCE METHOD

END OF YEAR	DEPRECIATION	CURRENT VALUE
1	1600	6400
2	1280	5120
3	1024	4096
4	819.2	3276.8
5	655.36	2621.44
6	524.288	2097.15
7	419.43	1677.72
8	335.544	1342.18
9	268.435	1073.74
10	214.748	858.993

TIME: 0.43 SECS.

(ب)

$$\begin{aligned} V &= 78000 \\ N &= 710 \\ I &= 71 \end{aligned}$$

STRAIGHT-LINE METHOD

END OF YEAR	DEPRECIATION	CURRENT VALUE
1	800	7200
2	800	6400
3	800	5600
4	800	4800
5	800	4000
6	800	3200
7	800	2400
8	800	1600
9	800	800
10	800	0

TIME: 0.45 SECS.

(١)

$$\begin{aligned} V &= 78000 \\ N &= 710 \\ I &= 73 \end{aligned}$$

SUM-OF-THE-YEARS'-DIGITS METHOD

END OF YEAR	DEPRECIATION	CURRENT VALUE
1	1454.55	6545.45
2	1309.09	5236.36
3	1163.64	4072.73
4	1018.18	3054.55
5	872.727	2181.82
6	727.273	1454.55
7	581.818	872.727
8	436.364	436.364
9	290.909	145.455
10	145.455	2.47955E-5

TIME: 0.31 SECS.

(ج)

شكل ٤ - ٦

تبين الأشكال ٤ - ٦ (أ) ، (ب) ، (ج) المخرجات التي تم الحصول عليها بطريقة الخط المستقيم وطريقة توازن الانحراف المزدوج وطريقة جمع أرقام السنوات بالترتيب . ونحن نحسب استهلاك عنصر قيمته المبدئية \$ 8000 في كل حالة وفترة حياته التي يستهلك فيها هي عشر سنوات .

لاحظ أن الطريقتين الأخيرتين تنتجان استهلاكاً سنوياً كبيراً خلال السنوات الأولى ، واستهلاكاً سنوياً صغيراً جداً في السنوات القليلة الأخيرة من حياة المنصر . ونرى أيضاً ، أن المنصر تكون قيمته صغراً عند نهاية عمره وذلك عند استخدام الطريقة الأولى والأخيرة ، ولكن تبقى له قيمة محددة عند استخدام طريقة توازن الانحراف المزدوج . (والرقم 2.47955E-5 بدلا من صفر في شكل ٤ - ٦) ناتج عن تقريب الأعداد) .

٤ - ٥ تكوين الحلقات التكرارية — جملة FOR-TO

BUILDING A LOOP—THE FOR-TO STATEMENT

لقد رأينا فعلاً أنه يمكننا بناء حلقات تكرارية في البيسك باستخدام جملة IF-THEN وجملة GO TO . ويكون ذلك مناسباً إذا لم يكن معلوماً مسبقاً كم عدد مرات التكرار . ومع ذلك ، فغالباً ما يكون لدينا معرفة مسبقة بعدد مرات تنفيذ الحلقة التكرارية . ويمكن بناء الحلقة التكرارية تحت هذه الظروف ببساطة وذلك باستخدام جملي FOR-TO و NEXT .

تحدد جملة FOR-TO عدد مرات تنفيذ الحلقة التكرارية . ويجب أن تكون أول جملة في الحلقة التكرارية وتتضمن جملة FOR-TO متغيراً عادياً (بدون دليل) ويجب أن يكون متغيراً رقمياً يسمى المتغير الجارى وتغير قيمته بعد كل مرة تنفذ فيها الحلقة التكرارية وتحدد عدد مرات التنفيذ بتوصيف قيمة مبدئية وقيمة نهائية للمتغير الجارى .

مثال ٤ - ١٠

تبين الجملة التالية جملة نموذجية لـ FOR-TO

50 FOR I=1 TO 10

والمتغير I في هذا المثال هو المتغير الجارى . وسوف تحدد قيمة I بواحد في أول مرة تنفذ بها الحلقة التكرارية . وسوف تزداد قيمة I بالوحدة في كل مرة تكرر فيها الحلقة التكرارية ، حتى تصل I إلى قيمتها النهائية وهي 10 في آخر مرة تنفذ فيها الحلقة التكرارية . وسوف ينتهي التنفيذ طالما تتعدى قيمة I القيمة النهائية وهي 10 . وبذلك فإن الحلقة التكرارية المعرفة أعلاه بجملة FOR-TO سوف تنفذ 10 مرات .

وسوف تزداد دائماً قيمة المتغير الجارى بالوحدة إذا لم تحتو جملة FOR-TO أى أوامر عكس ذلك . ومع ذلك ، فيمكننا زيادة المتغير الجارى بقيمة أخرى غير الواحد إذا رغبتنا في ذلك . ويمكننا عمل ذلك بإضافة عبارة STEP إلى جملة FOR-TO كما هو موضح في المثال التالي .

مثال ٤ - ١١

نفرض أننا نريد تنفيذ حلقة تكرارية 50 مرة ، والمطلوب زيادة المتغير الجارى بمقدار وحدتين . بعد كل تنفيذ متعاقب . يمكننا كتابة :

75 FOR J=1 TO 99 STEP 2

وبذلك تحدد قيمة المتغير الجارى J بالقيمة 1 أثناء المرور الأول والقيمة 3 أثناء المرور الثانى ، والقيمة 5 أثناء المرور الثالث ، . إلخ . حتى تأخذ J القيمة 99 أثناء المرور رقم 50 (الأخير) .

لا يلزم أن يتقيد المتغير الجارى بقيمة صحيحة موجبة ، بل يمكن أن يأخذ قيا سالبة أو عشرية إذا رغبتا في ذلك . علاوة على ذلك ، فيمكن للمتغير الجارى أن يتناقص لكل تنفيذ متعاقب للحلقة التكرارية . (ويمكن إنجاز ذلك بتوصيف كمية سالبة في عبارة STEP) وأخيراً يمكن التعبير عن القيمة المبدئية والنهائية وقيمة STEP التي تعطى للمتغير الجارى بمتغيرات أو صيغ رياضية بجانب الأرقام .

مثال ٤ - ١٢

مبين فيما يلي توضيحات للحمل FOR-TO صالحة :

```
80 FOR X=-1.5 TO 2.7 STEP 0.1
15 FOR I=N TO 0 STEP -1
55 FOR K=N1 TO N2 STEP N8
80 FOR F=A/2 TO (B+C)+2 STEP K+1
```

تسمح بعض نسخ البيسك باستخدام الكلمة BY بدلا من STEP .

مثال ٤ - ١٣

يمكن أيضاً كتابة جملة FOR-TO المبينة في مثال ٤ - ١١ في بعض نسخ بييسك .

```
75 FOR J=1 TO 99 BY 2
```

٤ - ٦ إنهاء حلقة تكرارية — جملة NEXT

CLOSING A LOOP—THE NEXT STATEMENT

وكما تبدأ دائماً الحلقة التكرارية بجملة FOR-TO ، فإنها دائماً تنتهي بجملة NEXT . وتتكون الحلقة التكرارية الكاملة من كل الحمل المتضمنة بين جملتي FOR-TO و NEXT .

تتكون جملة NEXT من رقم الجملة تتبعها الكلمة الدالة NEXT ، ثم يتبعها اسم المتغير الجارى . يجب أن يكون المتغير الجارى هو نفسه المتغير الجارى الذى يظهر في جملة FOR-TO المناظرة .

مثال ٤ - ١٤

فيما يلي هيكل حلقة تكرارية بنيت باستخدام جملتي FOR-TO و NEXT .

```
50 FOR I=1 TO 10
...
90 NEXT I
```

وسوف تتكون الحلقة التكرارية من كل الحمل ابتداء من الجملة رقم 50 إلى الجملة رقم 90 وسوف يتم تنفيذها 10 مرات .

يجب أن تحتفظ بعدة قواعد عند بناء حلقة تكرارية FOR-TO...NEXT هذه القواعد تتلخص فيما يلي :

١ - يمكن أن يظهر المتغير الجارى في أى جملة داخل الحلقة التكرارية ولكن لا يمكن أن تتغير قيمته .

٢ - إذا كانت القيمة المبدئية للمتغير الجارى مساوية للقيمة النهائية وقيمة حجم الخطوة ليست صفراً ، فإن الحلقة التكرارية سوف تنفذ مرة واحدة .

٣ - لن تنفذ الحلقة التكرارية إطلاقاً تحت الشروط الثلاثة التالية :

(١) إذا كانت قيم المتغير الجارى المبدئية والنهائية متساوية وكان حجم الخطوة مساوياً لصففر .

- (ب) إذا كانت القيمة النهائية للمتغير الجارى أقل من القيمة المبدئية وكان حجم الخطوة موجبا .
 (ج) إذا كانت القيمة النهائية للمتغير الجارى أكبر من القيمة الأصلية وكان حجم الخطوة سالبا .
 (عادة تحدث الشروط الموصوفة في القاعدتين ٢ و ٣ كخطأ غير مقصود) .

٤ - يمكن تحويل التحكم إلى خارج الحلقة التكرارية ولكن ليس إلى داخلها . (يمكن إنجاز تحويل التحكم إلى خارج الحلقة التكرارية بجملة GO TO أو جملة IF-THEN أو جملة ON-GO TO)

مثال ٤ - ١٥

ادرس هيكل الحلقة التكرارية المبينة فيما يلي :

```
120 FOR X=0 TO 0.5 STEP 0.01
...
165 LET Z=X-Y
170 IF Z>Z1 THEN 250
...
195 NEXT X
...
250 PRINT X,Y,Z
```

يوضح هذا المثال استخدام المتغير الجارى (X) بداخل الحلقة التكرارية (بالتحديد في الجملة 165) . ونرى أيضاً أن الجملة 170 تحول التحكم إلى خارج الحلقة التكرارية إذا كانت قيمة Z أكبر من القيمة Z1 . وأخيراً ، نرى المتغير الجارى (X) الذى يظهر في جملة NEXT هو نفسه المتغير الجارى في جملة FOR-TO كما هو مطلوب .

لاحظ أن الحمل بداخل الحلقة التكرارية ، بين جملتي FOR-TO و NEXT تم إدخالها قليلا لناحية اليمين ، ليس هذا مطلوب ولكن خبرة جيدة في البرمجة ، حيث تسمح زحزحة الجملة بداخل الحلقة التكرارية لسهولة التعرف عليها .

غالباً يستخدم هيكل الحلقة التكرارية FOR-TO...NEXT في أنواع عديدة ومختلفة من المسائل . فيما يلي مثال لمسألة نموذجية تتضمن استخدام مثل هذه الحلقة التكرارية .

مثال ٤ - ١٦ حساب متوسط بيانات تلوث الهواء Averaging of Air Pollution Data

يمكن التعبير عن مستوى تلوث الهواء بدلالة نوعية الهواء ، أى ، عدد الدقائق الموجودة في السنتيمتر المكعب من الهواء الملوث . وعلى ذلك كلما زادت قيمة نوعية الهواء ، كلما ارتفع مستوى التلوث . قياس نوعية الهواء يعمل عدة مرات في اليوم في معظم المدن الكبرى .

ولنفرض أننا أعطينا جدولاً يحتوي على عدد N من قياسات نوعية الهواء عند أوقات مختلفة خلال اليوم ، كما هو مبين في الجدول ٤ - ١ (لاحظ الوقت المعطى بدورات 24 ساعة أى 13.00 تشير إلى 1.00 مساءً . ولاحظ أيضاً أن $N = 20$ في هذا المثال) . نود حساب قيمة متوسطة للمتوسط الحسابي لنوعية الهواء لمدة بأكملها . ويمكن إنجاز ذلك أولاً بحساب قيمة متوسطة لكل فترة زمنية (أى الفترة الزمنية ما بين قراءتين متعاقبتين) ، ثم بعد ذلك حساب متوسط كامل مرجح من المتوسطات الفردية .

جدول ٤ - ١

الوقت	نوعية الهواء	الوقت	نوعية الهواء
0:00	12.2	13:00	46.6
2:00	12.5	14:00	43.1
4:00	11.9	15:00	39.2
6:00	13.5	16:00	44.7
7:00	22.4	17:00	62.9
8:00	31.4	18:00	88.0
9:00	57.7	19:00	71.4
10:00	84.4	20:00	59.0
11:00	68.0	22:00	43.5
12:00	51.6	24:00	28.7

طريقة إجراء الحسابات Computational Procedure

دعنا أولاً نقدم الرموز التالية :

T_1 = الوقت عند بدء الفترة الزمنية المعطاة .

T_2 = الوقت عند انتهاء الفترة الزمنية المعطاة .

Q_1 = نوعية الهواء عند بدء الفترة الزمنية المعطاة .

Q_2 = نوعية الهواء عند انتهاء الفترة الزمنية المعطاة .

Q_3 = متوسط نوعية الهواء خلال الفترة الزمنية المعطاة .

وسوف نجرى الحسابات كما يلي :

١ - حساب متوسط نوعية الهواء خلال كل فترة زمنية باستخدام الصيغة الرياضية :

$$Q_3 = (Q_1 + Q_2)/2$$

٢ - ضرب كل متوسط لنوعية الهواء بالفترة الزمنية المناظرة ($T_2 - T_1$) وتجميع كل حواصل الضرب ، أى :

$$S = [Q_3 * (T_2 - T_1)]_1 + [Q_3 * (T_2 - T_1)]_2 + \dots + [Q_3 * (T_2 - T_1)]_{N-1}$$

تشير S في الصيغة الرياضية السابقة إلى مجموع حواصل الضرب الفردية وتشير الأدلة 1 و 2 و ... و $N - 1$ إلى الفترات الزمنية العديدة . (لاحظ أن عدد الفترات الزمنية سوف تكون $N - 1$ ، حيث يوجد N قيمة لكل من Q و T) .

٣ - اقم هذا المجموع على الفترة الزمنية الشاملة (زمن آخر قراءة مطروح منه الزمن عند أول قراءة) وذلك للحصول على قيمة متوسط الوقت .

$$A = S / (T_9 - T_0)$$

حيث تشير A لمتوسط زمن نوعية الهواء ، وتشير كل من T_9 و T_0 إلى أول قراءة وزمن آخر قراءة بالترتيب .

(وسوف يتعرف القارئ الذى سبق ودرس حساباً عددياً على هذه المسألة كثرين أولى فى التكامل العدى . وتعرف الطريقة بقاعدة شبه المنحرف) .

المخطط التمهيدى للبرنامج The Program Outline

يمكننا الآن كتابة المخطط التمهيدى المفصل التالى لإجراء الحسابات .

١ - اقرأ N

٢ - اجعل قيمة S مساوية للصفر قبل قراءة أى من البيانات الحقيقية .

٣ - اقرأ أول مجموعة من البيانات (أى القيم المبدئية للتنبيرين $T1$ و $Q1$) .

٤ - اجعل قيمة $T0$ مساوية لقيمة $T1$ (هذه طريقة من طرق إعطاء علامة لزمن أول قراءة . سوف نحتاج هذه القيمة فى الخطوة السابعة التالية) .

٥ - نفذ الحسابات التالية عدد $(N - 1)$ من المرات :

(أ) اقرأ $T2$ و $Q2$

(ب) احسب $Q3$

(ج) اجمع حاصل ضرب $(T2 - T1) * Q3$ على قيمة S .

(د) اجعل قيمة $T2$ مساوية $T1$ و $Q2$ مساوية $Q1$ ، للتجهيز لفترة الزمن التالية . (بمعنى آخر ، تصبح قيم T و Q عند نهاية فترة زمنية معطاة هى قيم T و Q عند بداية الفترة الزمنية التالية) .

٦ - اجعل قيمة $T9$ مساوية لآخر قيمة $T2$.

٧ - احسب A .

٨ - اطبع A والفترة الزمنية الشاملة ، $(T9 - T0)$

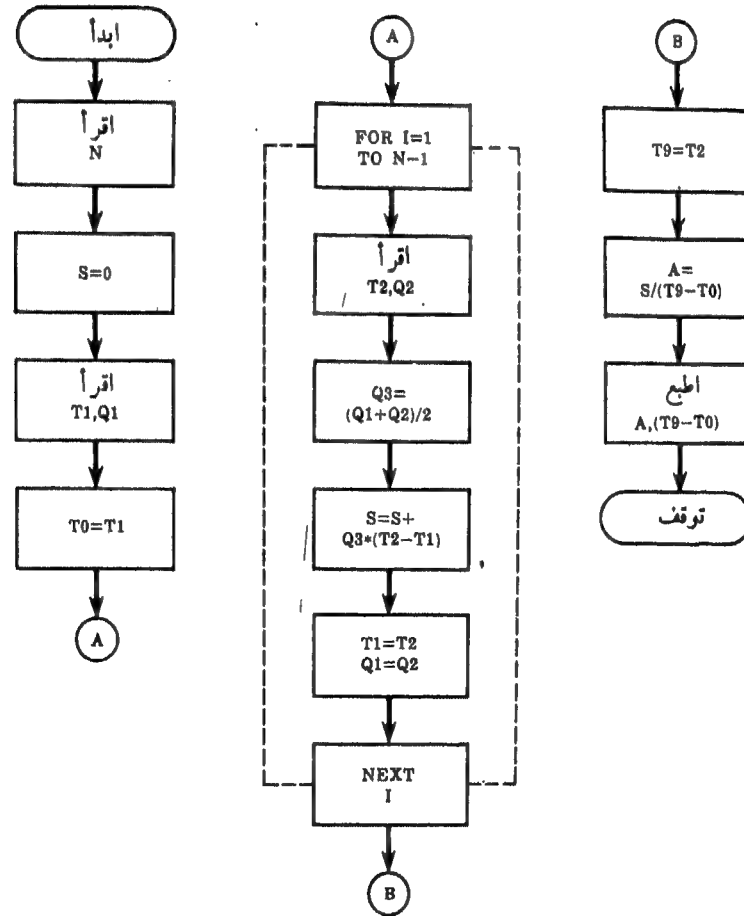
٩ - توقف .

يبين شكل ٤ - ٧ خريطة سير عمليات مناظرة للمخطط التمهيدى الأعلى . لاحظ أن الحلقة التكرارية FOR-TO...NEXT ثم حصرها بداخل مستطيل من الخطوط المتقطعة .

برنامج البيسك The Basic Program

يبين شكل ٤ - ٨ برنامج بيسك كاملا يناظر المخطط التمهيدى الممثل عالياً ، نرى أن المجموع التراكمى S يتكون بواسطة حلقة FOR-TO...NEXT التى تتكون من الجمل 100 إلى 160 . لاحظ أن البرنامج لا يتطلب تخزين القائمة الكاملة من بيانات الإدخال من أجل حساب هذا المجموع التراكمى (وسوف نرى طريقة مناسبة لتخزين قائمة كاملة ، إذا وجب علينا تنفيذ ذلك فى الفصل التالى) .

يبين الجزء الأدنى من شكل ٤ - ٨ قيم المدخلات والنتائج المحسوبة التى تناظر البيانات فى الجدول ٤ - ١ . نرى أن المتوسط الشامل لنوعية الهواء هى 41.5 لفترة 24 ساعة المعطاة . (لاحظ أن الإجابة لن تكون مضبوطة لأكثر من 3 أرقام معنوية ، حيث أن هذه هى دقة البيانات المدخلة) وسنجد أن المعلومات المدخلة موضوع بحثها خط .



شكل ٤ - ٧

٧ - ٤ الحلقات التكرارية المتداخلة NESTED LOOPS

يمكن تسكين حلقة تكرارية بداخل حلقة تكرارية أخرى (أى تداخل) . إذا رغبتنا ذلك . فى الحقيقة ، يمكن وجود عدة مستويات للتداخل وتطبق قواعد كتابة حلقة تكرارية مفردة أيضاً على الحلقات التكرارية المتداخلة . وبالإضافة لذلك يجب ملاحظة القيود التالية :

- ١ - يجب أن تبدأ كل حلقة متداخلة بجملة FOR-TO الخاصة بها وأن تنتهى بجملة NEXT الخاصة بها .
- ٢ - الحلقة التكرارية الخارجية والداخلية (المتداخلة) لا يمكن أن يكون لها نفس المتغير الجارى .
- ٣ - كل حلقة داخلية (متداخلة) يجب أن تسكن كاملاً بداخل الحلقة التكرارية الخارجية (أى لا يمكن للحلقات التكرارية أن تتشابك) .
- ٤ - يمكن تحويل التحكم من حلقة داخلية (متداخلة) إلى جملة فى الحلقة التكرارية الخارجية أو إلى جملة خارج نطاق الحلقات المتداخلة بأكملها . ومع ذلك ، لا يمكن تحويل التحكم إلى جملة داخل الحلقات المتداخلة من نقطة خارج مدى الحلقات المتداخلة .

```

10 REM AVERAGING OF AIR POLLUTION DATA
20 PRINT "N=";
30 INPUT N
40 LET S=0
50 PRINT
60 PRINT " T  Q"
70 PRINT
80 INPUT T1,Q1
90 LET T0=T1
100 FOR I=1 TO N-1
110   INPUT T2,Q2
120   LET Q3=(Q1+Q2)/2
130   LET S=S+Q3*(T2-T1)
140   LET T1=T2
150   LET Q1=Q2
160 NEXT I
170 LET T9=T2
180 LET A=S/(T9-T0)
190 PRINT
200 PRINT "AVERAGE AIR QUALITY=";A;"TIME INTERVAL=";T9-T0;"HOURS"
210 END

```

>RUN

EX4.16 21:58 14-MAR

N= 20

T Q

```

70, 12.2
72, 12.5
74, 11.9
76, 13.5
77, 22.4
78, 31.4
79, 57.7
710, 84.4
711, 88.0
712, 51.4
713, 44.6
714, 43.1
715, 39.2
716, 44.7
717, 62.9
718, 88.0
719, 71.4
720, 59.0
722, 45.5
724, 28.7

```

AVERAGE AIR QUALITY= 41.5354 TIME INTERVAL= 24 HOURS

TIME: 0.43 SECS.

شكل ٤ - ٨

مقال ٤ - ١٧

يبين الشكل التالي هيكل البرنامج يحتوي على حلقة تكرارية متداخلة.

```

100 FOR I=0 TO N STEP 2
...
120   FOR J=I TO N
...
160   NEXT J
...
200 NEXT I

```

لاحظ أن الحلقة الداخلية (الجمل 120 إلى 160) موجودة بالكامل داخل الحلقة الخارجية (الجمل 100 إلى 200) . كل حلقة تكرارية تبدأ وتنتهي بحملتي FOR-TO و NEXT الخاصة بها . وكل حلقة تكرارية لها المتغير الجارى الخاص بها . ومع ذلك ، لاحظ أن المتغير الجارى للحلقة الخارجية (I) قد استخدم كقيمة أولية للمتغير الجارى للحلقة الداخلية (J) وهذا مسموح به مادامت قيمة I لا تتغير بداخل الحلقة الداخلية .

تعطى الحلقات التكرارية وسائل مناسبة للقيام بعمل تكرار مجموعة من الحسابات . وإليك المثال التالى :

مثال ٤ - ١٨ توليد أرقام فيبوناتشى (Fibonacci) والبحث عن الأرقام الأولية :

المعروف أن أرقام فيبوناتشى (Fibonacci) أعضاء لتسلسل شيق حيث كل رقم يساوى مجموع الرقمين السابقين . بمعنى آخر .

$$F_i = F_{i-1} + F_{i-2}$$

حيث تشير F_i للرقم الذى يحمل الترتيب i . رقما فيبوناتشى (Fibonacci) الأوليان تم تعريفهما بالوحدة ١ أى

$$F_1 = 1$$

$$F_2 = 1$$

وعلى ذلك

$$F_3 = F_2 + F_1 = 1 + 1 = 2$$

$$F_4 = F_3 + F_2 = 2 + 1 = 3$$

$$F_5 = F_4 + F_3 = 3 + 2 = 5$$

وهكذا .

كل أعداد فيبوناتشى كميات صحيحة موجبة ، وبعض منها سوف تكون أولية . والرقم الأول الموجب هو رقم صحيح يقبل القسمة ، بدون باق ، على نفسه وعلى الواحد الصحيح فقط . فمثلا 5 رقم أول لأن الكميات التى يقبل القسمة عليها بدون باق هى 5 و 1 . وبمعنى آخر 8 ليست رقماً أولياً لأن 8 تقبل القسمة على 2 و 4 بجانب 1 و 8 .

طريقة إجراء الحسابات Computational Procedure

من السهل حساب أول N رقم من أرقام فيبوناتشى باستخدام الصيغة الرياضية أعلاه . ومع ذلك ، فإن الإجراء الذى يحدد ما إذا كان الرقم أولياً أو غير أولى يتطلب بعض الشرح .

ولنفرض أننا نريد تحديد أن رقماً معيناً قيمته أكبر من 2 يمكن قسمته بالتساوى على رقم صحيح أصغر . ولنسم الرقم الصحيح F والمقسوم عليه J وطريقة حساب خارج القسمة Q هى :

$$Q = F/J$$

وبعد ذلك نحسب خارج قسمة ممتوراً $Q1$ كما يلى :

$$Q1 = \text{INT}(Q)$$

وتشير الحروف INT لدالة يبسك المكتبية التى تحدد أكبر رقم صحيح لا يتعدى قيمة Q (وسوف نقول الكثير عن دوال يبسك المكتبية فى الفصل التالى) . وبذلك إذا كانت Q لها القيمة 5.3 ، فإن $Q1$ سيكون لها القيمة 5 .

وإذا كانت $Q1$ و Q لهما نفس القيمة ستكون F مقسومة بالتساوى على J . وعلاوة على ذلك ، إذا قسمت F بالتساوى على أى قيمة للمتغير J من $J = 2$ إلى $J = \text{INT} \sqrt{F}$ ، فإن F لا يمكن أن تكون أولية . حيث F ستكون رقماً أولياً فقط إذا كانت $Q1$ و Q لا تساوى أى قيمة من $\text{INT} \sqrt{F}$ ، 2، 3، ... J

الخطط التمهيدية للبرنامج The Program Outline

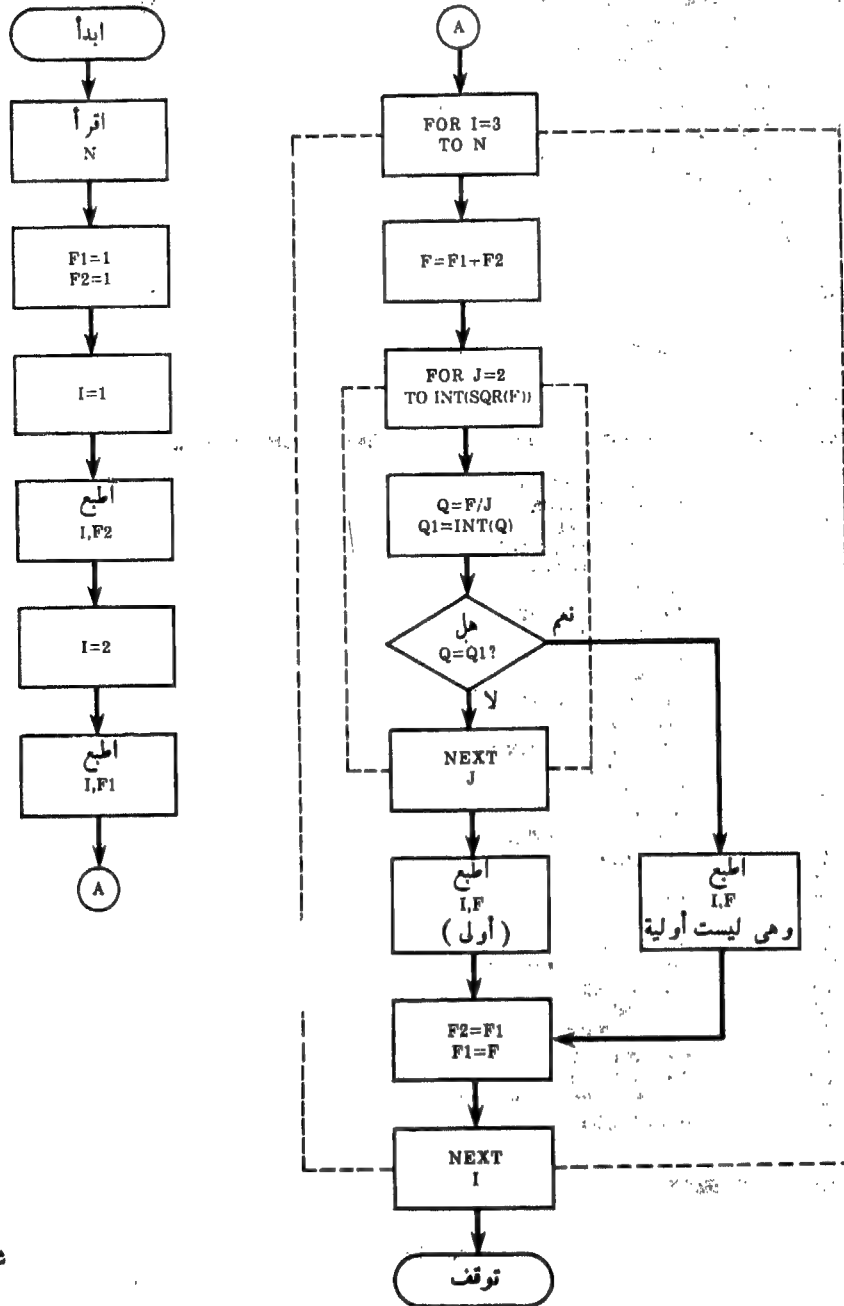
دعنا نشير إلى F كرقم فيبوناتشي (أي F_i) و $F1$ كرقم فيبوناتشي السابق (F_{i-1}) و $F2$ كرقم ثان سابق (F_{i-2}).
يمكننا الآن كتابة مخطط التمهيد للبرنامج الخاص بنا كما يلي :

١ - اقرأ N

٢ - اجعل $F1$ و $F2$ مساويين 1

٣ - اطبع $F1$ و $F2$ وتعرف على أن كلا منها رقم أول.

٤ - كرر نفس الحسابات للقيم $I = 3, 4, \dots, N$



شكل ٤ - ٩

```

10 REM GENERATION OF FIBONACCI NUMBERS AND SEARCH FOR PRIMES
20 PRINT "N=";
30 INPUT N
40 PRINT
50 PRINT "GENERATION OF FIBONACCI NUMBERS AND SEARCH FOR PRIMES"
60 PRINT
70 LET F1=1
80 LET F2=1
90 PRINT "I=";1,"F=";1; (PRIME)
100 PRINT "I=";2,"F=";1; (PRIME)
110 FOR I=3 TO N (GENERATE FIBONACCI NUMBERS)
120 LET F=F1+F2
130 FOR J=2 TO INT(SQR(F)) (TEST FOR A PRIME NUMBER)
140 LET G=F/J
150 LET Q1=INT(G)
160 IF G=Q1 THEN 200
170 NEXT J
180 PRINT "I=";I,"F=";F; (PRIME)
190 GOTO 210
200 PRINT "I=";I,"F=";F
210 LET F2=F1
220 LET F1=F
230 NEXT I
240 END

```

>RUN

EX4.18 09:26 15-MAR

N= 730

GENERATION OF FIBONACCI NUMBERS AND SEARCH FOR PRIMES

I= 1	F= 1 (PRIME)
I= 2	F= 1 (PRIME)
I= 3	F= 2 (PRIME)
I= 4	F= 3 (PRIME)
I= 5	F= 5 (PRIME)
I= 6	F= 8
I= 7	F= 13 (PRIME)
I= 8	F= 21
I= 9	F= 34
I= 10	F= 55
I= 11	F= 89 (PRIME)
I= 12	F= 144
I= 13	F= 233 (PRIME)
I= 14	F= 377
I= 15	F= 610
I= 16	F= 987
I= 17	F= 1597 (PRIME)
I= 18	F= 2584
I= 19	F= 4181
I= 20	F= 6765
I= 21	F= 10946
I= 22	F= 17711
I= 23	F= 28657 (PRIME)
I= 24	F= 46368
I= 25	F= 75025
I= 26	F= 121393
I= 27	F= 196418
I= 28	F= 317811
I= 29	F= 514229 (PRIME)
I= 30	F= 832040

TIME: 0.35 SECS.

(١) احسب قيمة للمتغير F باستخدام الصيغة الرياضية .

$$F = F1 + F2$$

(ب) افعل ذلك لقيم $J = 2, 3, \dots, R$ حيث R أكبر رقم صحيح لا يتعدى \sqrt{F} (اختبار الرقم الأول)

(١) احسب قيم للمتغيرات Q و Q1 واختبر ما إذا كانا متساويان .

(ii) إذا تساوت Q و Q1 لأى قيمة من J فإن F لا يمكن أن تكون رقماً أولياً . حيث تطبع I و F ، واصل مباشرة للخطوة ٤ (ج) التالية .

(iii) أما إذا كانت Q لا تساوى Q1 لكل قيم J فإن F يجب أن تكون أولية . حيث تطبع I و F ونعرف بأنها أولية ، واصل للخطوة ٤ (ج) التالية .

(ج) عدل قيمتي F1 و F2 (أى حدد القيمة الحالية F2 بالقيمة الحالية F1 ثم حدد قيمة F1 بالقيمة الحالية F) وذلك للتجهيز لحساب رقم فيبوناتسى جديد (أى قيمة جديدة للمتغير F) .

• - توقف

يبين شكل ٤ - ٩ خريطة سير عمليات مناظرة .

برنامج البيسك The BASIC Program

يبين شكل ٤ - ١٠ برنامج بيسك كاملاً مناظراً للمخطط التمهيدى السابق . لاحظ أن البرنامج يحتوى على تداخل فى الحلقات التكرارية . الغرض من الحلقة الداخلية (الجمل 130 إلى 170) هو تحديد ما إذا كان رقم فيبوناتسى أولياً أم لا ، بينما تسبب الحلقة الخارجية (الجمل 110 إلى 230) توليد أرقام فيبوناتسى المتتالية . لاحظ استخدام الدالة المكتنية SQR (فى السطر 130) ، وقد استخدمت للحصول على الجذر التربيعى للمتغير F (انظر القسم ٥ - ١) . ولاحظ أيضاً ، التحول المشروط للتحكم إلى خارج الحلقة الداخلية إذا كانت $Q = Q1$ (الجملة 160) .

يبين الجزء الأسفل من شكل ٤ - ١٠ المخرجات التى تم توليدها عند تنفيذ البرنامج للقيمة $N = 30$. نلاحظ أن 11 رقماً من الثلاثين رقماً الأولى لفيبوناتسى أرقاماً أولية . (لاحظ أن اجابات المستخدم موضوع تحتها خط) .

اسئلة للمراجعة

Review Questions

- ١ - ٤ هل اتخاذ قرار منطقي من الصفات الهامة للحاسبات الرقمية ؟ اشرح الأسباب لإجابتك .
- ٢ - ٤ ما هو المقصود بتحويل التحكم بداخل برنامج بيسك ؟
- ٣ - ٤ ما هي عملية التفرع المشروطة ؟ وكيف تختلف هذه العملية عن عملية التفرع غير المشروطة ؟
- ٤ - ٤ ما هي عملية التكرار ؟ وما هو الغرض من مثل هذه العملية ؟
- ٥ - ٤ اذكر أسماء المعاملات الست الرابطة المستخدمة في البيسك ؟ وما هو الغرض منها ؟
- ٦ - ٤ ما هو الغرض من جملة IF-THEN ؟
- ٧ - ٤ لخص قواعد كتابة جملة TF-THEN و اشرح ماذا يحدث عند تنفيذ هذه الجملة .
- ٨ - ٤ كيف تختلف جملة IF-THEN عن جملة GO TO ؟ وهل يمكن استخدام هاتين الجملتين سوياً للقيام بتنفيذ عملية منطقية مشتركة ؟
- ٩ - ٤ ما هو المقصود من إجراء تكرارى ؟
- ١٠ - ٤ ما هو الغرض من جملة ON-GO TO ؟
- ١١ - ٤ كيف تختلف جملة ON-GO TO عن جملة IF-THEN ؟
- ١٢ - ٤ لخص قواعد كتابة جملة ON-GO TO و اشرح ماذا يحدث عند تنفيذ هذه الجملة .
- ١٣ - ٤ ماذا يحدث بالضبط عند بتر الرقم ؟
- ١٤ - ٤ ما هو الغرض من جملة STOP ؟ وكيف تختلف عن جملة END ؟
- ١٥ - ٤ ما هو الغرض من جملة FOR-TO ؟ وما هو الغرض من المتغير الجارى وعبارة STEP ؟
- ١٦ - ٤ هل يمكن أن يأخذ المتغير الجارى قيمة عشرية (كسر) أو قيمة سالبة ؟
- ١٧ - ٤ هل يمكن إنقاص قيمة المتغير الجارى عند تنفيذ الحلقة التكرارية مرات متتالية ؟
- ١٨ - ٤ ما هو الغرض من جملة NEXT ؟ ما هو المطلوب من المتغير الجارى الذى يظهر فى هذه الجملة ؟
- ١٩ - ٤ اذكر طريقتين مختلفتين يمكن أن تكون بهما الحلقة التكرارية في البيسك و اذكر نوع الموقف الذى يتناسب مع كل نوع من أنواع الحلقات التكرارية ؟
- ٢٠ - ٤ هل يمكن للمتغير الجارى أن يظهر فى جملة من الجمل التى تحتويها الحلقة التكرارية بداخل FOR-TO.....NEXT ؟ وهل يمكن تغيير قيمة المتغير الجارى فى مثل هذه الجملة ؟

- ٢١-٤ تحت أى ظروف سوف تنفذ حلقة FOR-TO.....NEXT مرة واحدة ؟ أو لا تنفذ على الإطلاق ؟
- ٢٢-٤ ما هى القيود التى تطبق على تحويل التحكم من أو إلى حلقة FOR-TO.....NEXT ؟
- ٢٣-٤ لماذا تظهر الجمل فى حلقة FOR-TO.....NEXT دائماً مزخرفة إلى الداخل ؟ وهل هذه الزخرفة مهمة ؟
- ٢٤-٤ ما هو المقصود من الحلقات التكرارية المتداخلة ؟
- ٢٥-٤ لحسن القواعد التى تطبق لتداخل الحلقات التكرارية FOR-TO.....NEXT قارن ذلك بالقواعد التى تطبق على حلقة تكرارية مفردة .

مسائل محلولة

Solved Problems

- ٢٦-٤ كل شرط من الشروط التالية يتضمن استخدام المعاملات الرباعية . تعرف على أى من هذه الشروط . مكتوب بطريقة غير صحيحة إن وجد .

الخطأ	الشرط
لا يمكن مقارنة متغير رقمى مع متغير حرفى . صحيح	X="DATE" K+2>=100
لا يمكن مقارنة متغير حرفى بقيمة عددية . صحيح	N\$<>A+B P\$="123456"
لا يسمح بالصيغ الرياضية التى تحتوى حروفاً أو متغيرات حرفية .	T\$=R\$+S\$

- ٢٧-٤ مبين فيما يلى عدة جمل IF-THEN حدد ، إذا كانت أى منها قد كتبت بطريقة غير صحيحة .

الخطأ	الجملة
صحيحة	20 IF K+2>=100 THEN 50
صحيحة	20 IF (K+2>=100) THEN 50
ليست كل نسخ بيسك (BASIC) تسمح باستخدام GO TO بدلاً من THEN	20 IF K+2>=100 GO TO 50
يجب أن يكون رقم الجملة التى يتحول إليها التحكم رقماً صحيحاً موجباً وليس متغيراً .	65 IF X+Y<>Z THEN M
صحيحة	100 IF G\$="MAY 13" THEN 45
بناء هيكل غير صحيح لغوياً (يجب أن تبدأ الجملة بالكلمة IF) .	35 GO TO 150 IF J=3
بناء هيكل غير صحيح لغوياً (يجب أن تتبع THEN رقم جملة فقط) .	50 IF X1<=50 THEN X=X+5

٤ - ٢٨ مبين فيما يلي البناء الهيكل لعدة جمل IF-THEN....GO TO والى تكون حلقات تكرارية . حدد ، إذا كانت أى منها قد كتبت بطريقة غير صحيحة .

```

20...
...
60 IF N>N1 THEN 110
...
85 LET N=N+1
90 GO TO 20
...
110...

```

(أ)

```

35 LET P=0
...
50...
...
75 IF P<=0 THEN 125
80 GO TO 50
...
125...

```

(ب)

يتحول التحكم دائماً إلى الجملة رقم 125 ، حيث تم تحديد قيمة مبدئية للمتغير P بصفر ولن تتغير قيمتها فيما بعد .

```

15...
...
45 INPUT X
50 IF X<50 THEN 90
...
85 STOP
90...
...
110 GO TO 15

```

(ج)

```

110 C=5
...
130...
...
160 IF C<=0 THEN 200
170 C=C+5
180 GO TO 180
...
200...

```

(د)

سوف تنتصر الحلقة التكرارية إلى ما لا نهاية حيث لن تكون قيمة C أقل أو تساوى صفر إطلاقاً .

٤ - ٢٩ مبين فيما يلي عدة جمل ON-GO TO حدد ، إذا كانت أى منها قد كتبت بطريقة غير صحيحة .

الجملة	الخطأ
15 ON X3 GO TO 25,15,25,40	صحيحة
100 ON 2*(C1+C2)/N GO TO K1,K2,K3	يجب أن تكون أرقام الجمل التي يتحول إليها التحكم أرقاماً صحيحة موجبة . وليست متغيرات
55 ON J+K THEN 120,90,150	لا تسمح كل نسخ بيسك باستخدام THEN بدلا من GO TO
80 ON T GO TO 25	يجب أن تملأ أرقام جملتين مختلفتين على الأقل .
20 ON N\$ GO TO 50,70,50,90	لا يمكن لتغير حرفي أن يظهر في جملة ON-GO TO
60 ON P(I) GO TO 10,120	صحيحة (لاحظ أن P(I) متغير ذو دليل) وسوف نناقش المتغيرات ذات الأدلة في الفصل الخامس .

٤ - ٣٠ مبين فيما يلي عدة جمل FOR-TO حدد ، إذا كانت أى منها قد كتبت بطريقة غير صحيحة .

الجملة	الخطأ
20 FOR N=J TO 3*(K+1) STEP J1	صحيحة
50 FOR N=0 TO -100 STEP -5	صحيحة
75 FOR X(I)=1 TO 100	لا يمكن استخدام متغير ذي دليل كتغير جار .
35 FOR K=K1 TO 100 STEP K1	صحيحة
65 FOR N\$=1 TO 19 STEP 2	لا يمكن استخدام متغير حرفي كتغير جار .
100 FOR I=J TO K STEP I	لا يمكن أن يظهر المتغير الجارى في عبارة STEP

٤ - ٣١ مبين فيما يلي عدة هياكل للحلقات التكرارية FOR-TO.....NEXT حدد إذا كانت أى منها ، مكتوبة بطريقة

غير صحيحة

20 FOR I=1 TO 100 STEP J

(١)

80 NEXT J

المتغير الجارى الموجود في جملة NEXT وهو (J) . ليس هو نفس المتغير الجارى في جملة FOR-TO وهو (I) .

50 FOR N=N1 TO N2

(ب)

...

75...

...

90 NEXT N

...

120 IF N=10 THEN 75

لا يمكن أن يتحول التحكم إلى داخل الحلقة التكرارية .

```

100 FOR K=3 TO -3 STEP -1
...
130 PRINT X↑K
...
150 NEXT K

```

(ج)

صحيفة ، بفرض أنه تم تحديد قيمة للمتغير X . (لاحظ ظهور المتغير الجارى بداخل جملة PRINT ، ولكن قيمته لم تتغير) .

```

25 FOR X=0 TO 1 STEP 0.05
...
50   FOR Y=0 TO 10 STEP 0.1
...
75   NEXT X
...
100 NEXT Y

```

(د)

تشابه الحلقة التكرارية الداخلية والحلقة التكرارية الخارجية .

```

100 FOR I=1 TO M
...
125   FOR J=1 TO N
...
135     FOR K=1 TO M+N
...
160     NEXT K
...
180   NEXT J
...
200   FOR J=1 TO N
...
225   NEXT J
...
235   FOR K=1 TO M+N
...
240   NEXT K
...
250 NEXT I

```

(هـ)

صحيفة

مسائل تكميلية

Supplementary Problems

٣٢ - ٤ يتضمن كل شرط مما يلي استخدام المعاملات الرابطة . حدد إذا كانت أي منها قد كتبت بطريقة غير صحيحة .

$X < .01$	(د)	$J \geq J1 + J2$	(ا)
$P2 < > T\$$	(هـ)	$C = C + 1$	(ب)
$A/B \leq C/D$	(و)	$N\$ = "END"$	(ج)

٣٣ - ٤ باهر المطلوب من أجل أن يتحقق الشرط :

$$P\$ < Q\$$$

في كل من السلاسل الحرفية المثلة بالمتغيرات $P\$$ و $Q\$$ ؟

٣٤ - ٤ ميين فيما يلي عدة جمل IF-THEN حدد إذا كانت أي منها قد كتبت بطريقة غير صحيحة .

150 IF P=P1 THEN K	(ا)	30 IF K < > K1 THEN 10	(ا)
100 IF B+2 < > (4*A*C) THEN 150	(و)	50 IF J < 100 THEN J=J+1	(ب)
45 IF X+2 < 0 THEN 20	(ز)	120 IF X > = Y+Z GO TO 200	(ج)
		75 IF M\$ = "DATE" THEN 50	(د)

٣٥ - ٤ اكتب جملة بيسك أو أكثر لتوافق كل موقف من المواقف الموصوفة التالية :

- (ا) حول التحكم للجملة رقم 50 إذا كانت قيمة K أقل من 15 وإلا نفذ الجملة التالية .
- (ب) حول التحكم للجملة رقم 70 إذا كانت قيمة N\$ مساوية للسلسلة "OPTION A" وإلا حول التحكم إلى الجملة رقم 150 .
- (ج) حول التحكم للجملة رقم 200 إذا كانت قيمة X أقل من أو تساوى 100 ؛ وإلا أضف مقدار الوحدة على المتغير J ، واقرأ قيمة جديدة للمتغير X ، ثم ارجع مرة ثانية للجملة 60 .
- (د) حول التحكم للجملة رقم 150 إذا كانت قيمة J مساوية للصفر ، وإلا أضف قيمة J على القيمة الموجودة في S ، ثم ارجع مرة ثانية للجملة رقم 20 .

٣٦ - ٤ ميين فيما يلي البناء الهيكل لعدة جمل IF-THEN.....GO TO والتي تكون حلقات تكرارية . حدد إذا كانت أي منها قد كتبت بطريقة غير صحيحة .

10 LET X=100 ... 50... ... 100 IF X>100 THEN 200 110 LET X=X-5 120 GO TO 50 ... 200...	(د)	20... ... 120 IF K\$="END" THEN 200 130 PRINT A,B,C 140 GO TO 20 ... 200 STOP	(ا)
10 LET X=100 ... 50... ... 100 IF X=0 THEN 200 110 LET X=X-5 120 GO TO 50 ... 200...	(هـ)	10 INPUT N 20 LET T=0 30 LET J=1 40 INPUT T1 50 LET T=T+T1 60 IF J=N THEN 150 70 LET J=J+1 80 GO TO 30 ... 150 PRINT J,T 160 END	(ب)
		10 LET X=100 ... 50... ... 100 IF X>=100 THEN 200 110 LET X=X-5 120 GO TO 50 ... 200...	(ج)

٤ - ٣٧ مبين فيما يلي عدة جمل ON-GO TO . حدد إذا كانت أي منها قد كتبت بطريقة غير صحيحة .

150 ON ((A+B)/C)+2 GO TO 170,190,225	(هـ)	50 ON N\$ GO TO 10,70,120	(ا)
45 ON K-3 THEN 65,90	(و)	100 ON A GO TO 50,20,50,200,120	(ب)
80 GO TO 50,120,150 ON (X+Y)	(ز)	75 ON X1 GO TO 100,25,75,150	(ج)
		100 ON N GO TO N1,N2,N3,N4,N5	(د)

٤ - ٣٨ ادرس الجملة ON-GO TO التالية (وهي صحيحة لغوياً) :

75 ON J-K GO TO 100,50,20,150

ماذا يحدث عند تنفيذ هذه الجملة إذا كانت :

J=5 and K=0?	(د)	J=2 and K=3?	(ا)
J=-1 and K=-2.8?	(هـ)	J=4.5 and K=0.75?	(ب)
		J=1 and K=-1?	(ج)

٤ - ٣٩ مبين فيما يلي عدة جمل FOR-TO . حدد إذا كانت أي منها قد كتبت بطريقة غير صحيحة .

125 FOR X=V(1) TO V(2) STEP V(3)	(د)	100 FOR C=.1*A TO .25*(A+B) STEP P/2	(ا)
30 FOR J1=25 TO -25 STEP -5	(هـ)	65 FOR X+2=0 TO 100 STEP 20	(ب)
45 FOR A=12 TO 0 STEP 0.1	(و)	80 FOR K\$=P\$ TO Q\$	(ج)

٤ - ٤ اكتب زوجاً مناسباً من جمل FOR-TO و NEXT (أى اكتب هيكل حلقة تكرارية FOR-TO.....NEXT لكل من المواقف الموصوفة التالية :

- (أ) حلقة تكرارية يجب تكرارها 200 مرة .
 (ب) حلقة تكرارية يجب تكرارها 200 مرة . ماعدا أن التحكم سوف يتحول خارج الحلقة إلى جملة 175 إذا أصبحت قيمة X أقل من 0.001
 (ج) حلقة تكرارية تكرر عدداً مناسباً من المرات للمتغير الجارى الذى يتزايد من 1 إلى 73 ، بفرض أن المتغير الجارى يزداد في كل مرة بمقدار 3 وحدات عند تنفيذ الحلقة التكرارية ،
 (د) حلقة تكرارية تكرر عدداً مناسباً من المرات للمتغير الجارى الذى يتزايد من 0.5 إلى قيمة معطاة بالصيغة الرياضية $10 - 3 \uparrow A$. في كل مرة تنفذ فيها الحلقة التكرارية تزداد قيمة المتغير الجارى بالقيمة المعطاة في الصيغة الرياضية $A + B$.

٤ - ٤١ مبين فيما يلى هياكل عدة حلقات تكرارية FOR-TO.....NEXT ، حدد إذا كانت أى منها قد كتبت بطريقة غير صحيحة :

100 FOR X=0 TO 1 STEP .02 (د)	10 FOR K=K1 TO K2 (أ)
... 150 FOR X=0 TO 5 STEP .05	... 50 K=K+1
... 200 NEXT X	... 90 NEXT K
50 FOR P=1 TO 50 (هـ)	10 FOR C1=0 TO 50 STEP 5 (ب)
... 80 FOR Q=2 TO 100 STEP 2	... 35 FOR C2=0 TO C1
... 100 IF T>=T1 TH 160	... 65 NEXT C2
... 120 NEXT Q	... 100 NEXT C1
... 160 PRINT "T=";T	10 FOR J=1 TO N (ج)
... 200 NEXT P	... 40 FOR K=1 TO M
75 FOR X=A TO (A+B) STEP C (و)	... 70 NEXT J
... 125 NEXT C	... 120 NEXT K

مسائل للبرمجة

Programming Problems

٤ - ٤٢ يمكن إعادة تنظيم المعادلة

$$x^5 + 3x^2 - 10 = 0$$

المطلبة في المثال ٤ - هـ بالصورة التالية :

$$x = \sqrt{(10 - x^5)/3}$$

أعد كتابة برنامج البيسك المروض في مثال ٤ - هـ حتى يتسنى له استخدام الصيغة السابقة للمعادلة . نفذ البرنامج مع طباعة قيم x المحسوبة أثناء كل تكرار . قارن النتائج المحسوبة مع النتائج المروضة في مثال ٤ - هـ .

٤-٣ أعد كتابة البرنامج المعروض في مثال ٥ - ٤ حتى يتسنى له استخدام الحلقة التكرارية FOR-TO....NEXT فضلا عن الحلقة التكرارية IF-THEN.....GO TO .

٤-٤ أعد كتابة البرنامج المعروض في مثال ٤ - ٩ وذلك لاستبدال جمل ON-GO TO بجمل IF-THEN. ما هي الفائدة التي تم الحصول عليها باستخدام جمل ON-GO TO ؟

٤-٥ أعد كتابة البرنامج المعروض في مثال ٤ - ١٦ حتى يتسنى له استخدام الحلقة التكرارية IF-THEN.....GO TO فضلا عن الحلقة التكرارية FOR-TO....NEXT

٤-٦ اكتب برنامجاً من النوع التخطي الذي يقرأ رقماً صحيحاً موجباً ويحدد :

(أ) ما إذا كان الرقم الصحيح رقماً أولياً .

(ب) ما إذا كان الرقم الصحيح ضمن أرقام فيبوناتشي (Fibonacci).

اكتب البرنامج بطريقة تمكنه من تكرار التنفيذ (حلقة تكرارية) ، حتى تكون الكمية المدخلة مساوية صفراً .

٤-٧ نعرض فيما يلي عدة تمارين بسيطة للبرمجة . جهز مخططاً تمهيدياً مفصلاً وخريطة سير عمليات مناظرة وبرنامج بينك كاملاً لكل من هذه التمارين . نفذ البرنامج باستخدام البيانات المعطاة .

(أ) احسب المتوسط الحسابي لقائمة من عدد n من الأرقام . اختر البرنامج الخاص بك بمجموعة البيانات التالية ($n = 10$)

27.5	87.0
13.4	39.9
53.8	47.7
29.2	8.1
74.5	63.2

(ب) احسب المتوسط الحسابي المرجح لقائمة تتكون من عدد n من الأرقام باستخدام الصيغة الرياضية :

$$x_{avg} = f_1x_1 + f_2x_2 + \dots + f_nx_n$$

حيث قيم f_i هي معاملات مرجحة كسرية أي أن $0 \leq f_i \leq 1$ و $f_1 + f_2 + \dots + f_n = 1$. اختر البرنامج الخاص بك باستخدام البيانات المعطاة في الجزء (أ) والمعاملات المرجحة التالية ($n = 10$)

$i = 1$	$f = 0.06$	$i = 6$	$f = 0.10$
2	0.08	7	0.12
3	0.08	8	0.12
4	0.10	9	0.12
5	0.10	10	0.12

(ج) احسب حاصل الضرب التراكمي لقائمة n من الأرقام . اختر البرنامج الخاص بك بالمجموعة التالية من البيانات ($n = 6$) :

6.2 و 12.3 و 5.0 و 18.8 و 7.1 و 12.8

(د) احسب المتوسط الهندسي لقائمة من الأرقام باستخدام الصيغة الرياضية :

$$x_{avg} = [x_1x_2x_3 \dots x_n]^{1/n}$$

اختر البرنامج الخاص بك باستخدام البيانات المعطاة في الجزء (ج) عاليه . قارن بين النتائج التي تم الحصول عليها من المتوسط الحسابي لنفس البيانات . أي من المتوسطات أكبر ؟

٤-٤٨ جهاز مخططاً تمهيدياً مفصلاً ، وخريطة سير عمليات مناظرة وبرنامج ببسك كاملاً لكل من المسائل المعروضة التالية :

(أ) احسب مجموع أول 100 رقم صحيح فردى أى (1 + 3 + 5 + + 199) اكتب البرنامج بطريقتين مختلفتين :

(i) اكتب مجموعة متتالية من الجمل والتي تجمع الرقم الصحيح التالى على المجموع فى كل مرة يعاد فيها التسلسل .

دع القرار لتكرار التسلسل أو للإنتهاء متوقفاً على ناتج جملة IF-THEN

(ii) استخدم حلقة تكرارية FOR-TO..... NEXT

ما هى المزايا والعيوب ؟ صف خصائص كل طريقة .

(ب) احسب مجموع حاصل ضرب عدد N مضروباً فى رقم صحيح K . (فثلاً ، إذا كانت K = 3 و N = 10

فاحسب مجموع 10 × 3 + + 2 × 3 + 1 × 3) اجعل البرنامج عاماً تماماً وذلك بقراءة قيم

N و K فى كل مرة ينفذ فيها البرنامج . اختر البرنامج بحساب مجموع أول 1000 من مضاعفات الرقم الصحيح 3 .

(ج) احسب قيمة K! ، حيث K تمثل رقماً صحيحاً تقرأ قيمته للحاسب كل مرة ينفذ فيها البرنامج . اختر البرنامج

بحساب قيمة 10! (لاحظ أن K! تعرف بـ K × 4... × 3 × 2 × 1)

(د) يمكن حساب جيب الزاوية x تقريباً وذلك بتجميع أول N حد من حدود المتوالية اللانهائية :

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (1)$$

اكتب برنامج ببسك يقرأ قيمة للمتغير x وبعد ذلك يحسب الجيب الخاص به . اكتب البرنامج بطريقتين مختلفتين .

(i) اجمع أول N من الحدود ، حيث تمثل N رقماً صحيحاً موجباً يقرأ فى الحاسب مع قيمة عددية للمتغير x .

(ii) استمر فى إضافة الحدود المتعاقبة فى المتوالية حتى تصبح قيمة الحد أصغر فى المقدار (من 10^{-5}) .

اختر البرنامج للقيم $x = 1$ و $x = 2$ و $x = 3$ وفى كل حالة اكتب عدد الحدود المستخدمة بجانب الإجابة النهائية .

(هـ) نفرض أنك وضعت كمية معينة من النقود A فى حساب ادخار عند بداية كل سنة لمدة n من السنوات . فإذا كان

الحساب يربح بسعر فائدة مئوية i سنوياً ، وبذلك فإن كمية النقود التى تم تراكمها بعد n من السنوات (F) معطاة بالصيغة :

$$F = A [(1+i/100) + (1+i/100)^2 + (1+i/100)^3 + \dots + (1+i/100)^n]$$

اكتب برنامج ببسك من النوع التخطي ليحدد ما يلى :

(i) كم تكون كمية النقود التى تتراكم بعد 30 سنة إذا كان المبلغ المودع \$1000 عند بداية كل سنة ونسبة الربح هى 6% سنوياً (حساب مركب سنوياً) ؟

(ii) ما هو المبلغ الذى يجب أن يودع عند بداية كل سنة حتى يتراكم إلى \$100,000 بعد 30 سنة (مرة أخرى نفترض أن نسبة الربح هى 6% سنوياً ، (مركبة سنوياً) ؟

في كل حالة ، ابدأ أولاً بتحديد المبلغ غير المعروف . ثم بعد ذلك أنشئ جدولاً مبيناً إجمال المبلغ الذي تم تراكمه عند نهاية كل سنة

(و) عدل البرنامج السابق ليتلاءم مع حساب الأرباح المركبة ربع سنوية وليس سنوياً . قارن النتائج المحسوبة بالنتائج التي تم الحصول عليها في الجزء (هـ) . تلميح الصيغة الرياضية الصحيحة هي :

$$F = A [(1+i/100m)^m + (1+i/100m)^{2m} + (1+i/100m)^{3m} + \dots + (1+i/100m)^{nm}]$$

حيث تمثل m عدد فترات الأرباح لكل سنة .

(ز) حصل فصل من الطلبة على الدرجات التالية لإمتحانات السنة المأخوذة في مقرر لغة البرمجة ببسك .

الاسم	درجات الاختبارات (نسبة مئوية)					
Adams	45	80	80	95	55	75
Brown	60	50	70	75	55	80
Davis	40	30	10	45	60	55
Fisher	0	5	5	0	10	5
Hamilton	90	85	100	95	90	90
Jones	95	90	80	95	85	80
Ludwig	35	50	55	65	45	70
Osborne	75	60	75	60	70	80
Prince	85	75	60	85	90	100
Richards	50	60	50	35	65	70
Smith	70	60	75	70	55	75
Thomas	10	25	35	20	30	10
Wolfe	25	40	65	75	85	95
Zorba	65	80	70	100	60	95

اكتب برنامج ببسك بصيغة تخاطبية يقبل اسم الطالب ودرجاته كمدخلات ويحسب متوسط الدرجات لكل طالب . المطلوب أن يكون البرنامج عاماً بقدر الإمكان .

(ح) عدل البرنامج المكتوب في جزء (ز) السابق ليسمح بأوزان غير متساوية لدرجات الاختبار الفردية . عموماً ، افرض أن كلا من الامتحانات الأربعة الأولى تساهم بمقدار 15% للدرجة النهائية ، وأن كلا من الامتحانين الآخرين تساهم بمقدار 20% .

(ط) أضف إلى البرنامج المكتوب في الجزء (ح) السابق حتى يتسنى حساب متوسط شامل للفصل بالإضافة للمتوسطات الفردية للطلبة .

(ي) اكتب برنامج ببسك يسمح باستعمال النهاية الطرفية المركزية كآلة حاسبة مركزية . أخذاً في الاعتبار العمليات الحسابية الشائعة فقط (الجمع والطرح والضرب والقسمة) .

(ك) اكتب برنامج ببسك لحساب جذور المعادلة التربيعية :

$$ax^2 + bx + c = 0$$

(انظر الأمثلة ٢ - ٢٦ و ٢ - ٣٠) اصحح بإمكانية جعل أحد الثوابت له القيمة صفر ، وأن الكمية $b^2 - 4ac$ تكون أقل من أو تساوى صفراً .

الفصل ٥

بعض الخصائص الإضافية شائعة الاستعمال للغة البيسك Some Additional Features of BASIC

يمرض هذا الفصل بعض الخصائص الإضافية شائعة الاستعمال للغة البيسك . وسنبداً بمناقشة الدوال المكتبية المبنية داخلياً والتي تبسط بعض العمليات الحسابية الشائعة مثل حساب القيمة المطلقة لرقم ولوغاريتم عدد و . . . الخ . ثم بعد ذلك سوف ندرس استخدام القوائم والجداول التي تسمح لنا بتداول مجموعة من الكميات العددية أو الحرفية كما لو كانت متغيرات فردية . وأخيراً سوف نعرض جملتين إضافيتين لإدخال البيانات ، وبذلك ، تممنا هاتان الجملتان بطريقة بديلة لاستخدام جملة INPUT .

٥ - ١ الدوال المكتبية LIBRARY FUNCTIONS

تممنا الدوال المكتبية للغة البيسك (وتسمى أيضاً دوال عامة أو دوال أولية) بطريقة سريعة ومهلة لحساب عدة دوال حسابية وللقيام بعمليات منطقية معينة . هذه الدوال المكتبية عبارة عن برامج صغيرة فرعية سبق كتابتها وتعتبر كجزء متكامل من اللغة . يمكن التوصل لأي دالة ببساطة بذكر اسمها ثم يل الاسم أى معلومات يجب أن تعطى للدالة محصورة بين قوسين . (الكمية العددية أو سلسلة الحروف التي تصل للدالة بهذه الطريقة تسمى خلاصة) . وفور التوصل إلى الدالة فسوف تنفذ العملية الحسابية المطلوبة أو توماتيكياً بدون الحاجة لكتابة برنامج مفصل لها .

مثال ٥ - ١

نفرض أن المطلوب حساب الجذر التربيعي للقيمة الممثلة بالمتغير X . فيمكننا كتابة :

```
50 LET Y=SQR(X)
```

وذلك يسبب تحديد قيمة Y بقيمة الجذر التربيعي للمتغير X. وتسمى الدالة التي تحسب الجذر التربيعي SQR ، والخلاصة في هذا المثال هو المتغير X .

كان من الممكن بالطبع كتابة :

```
50 LET Y=X↑.5
```

وبذلك فاستخدام دالة الجذر التربيعي غير مطلوب . ولكنه مجرد شيء مفيد . (ويجب الإشارة مع ذلك ، إلى أن حساب الجذر التربيعي للرقم باستخدام دالة الجذر التربيعي تستغرق من وقت الحاسب كمية أقل من عملية الأس المناظرة) .

يمثل جدول ٥ - ١ عدة دوال مكتبية شائعة . وهناك قائمة أكثر مبينة في الملحق ب ، وتتضمن أيضاً المداخل الموجودة في جدول

٥ - ١ .

جدول ٥ - ١ دوال مكتوبة شائعة الاستخدام

الدالة	التطبيق	الوصف
ABS	10 LET Y=ABS(X)	تحسب القيمة المطلقة للمتغير x ، $y = x $
ATN	10 LET Y=ATN(X)	تحسب مقابل الظل للمتغير x ، $y = \arctan(x)$
COS	10 LET Y=COS(X)	تحسب جيب تمام المتغير x ، $y = \cos(x)$ حيث x بالتقدير الدائري
COT	10 LET Y=COT(X)	تحسب ظل تمام المتغير x ، $y = \cot(x)$ حيث x بالتقدير الدائري
EXP	10 LET Y=EXP(X)	ترفع e للقوة x ؛ $y = e^x$
INT	10 LET Y=INT(X)	تحديد قيمة y بأكبر رقم صحيح حيث لا يتعدى قيمة x
LOG	10 LET Y=LOG(X)	تحسب قيمة اللوغاريتم الطبيعي للمتغير x ، $y = \log_e x$ و $x > 0$
SGN	10 LET Y=SGN(X)	تحدد إشارة المتغير x ($y = +1$ إذا كانت x موجبة و $y = 0$ إذا كانت $x = 0$ و $y = -1$ إذا كانت x سالبة)
SIN	10 LET Y=SIN(X)	تحسب جيب المتغير x ، $y = \sin(x)$ ، حيث x بالتقدير الدائري
SQR	10 LET Y=SQR(X)	تحسب الجذر التربيعي للمتغير x ، $y = \sqrt{x}$ ، حيث $x > 0$
TAB	20 PRINT TAB(N);X	تسبب تحريك رأس الطباعة للنهاية الطرفية للكتابة لتكون في موضع معين . (العمود الأيسر يعتبر العمود رقم صفر)
TAN	10 LET Y=TAN(X)	تحسب الظل للمتغير x ، $y = \tan(x)$ حيث x بالتقدير الدائري

$+$ يمثل الرمز e أساس النظام الطبيعي للوغاريتمات (Naperian) وهو رقم غير منطقي . وتساوى قيمته التقريبية 2.718282

استخدام كل الدوال الموجودة في الجدول ٥ - ١ يجب أن يكون واضحاً ما عدا دالة INT ودالة TAB وسوف نناقشها فيما بعد . لاحظ أن بعض الدوال (مثل LOG ، SQR) تتطلب خلاصة موجبة . إذا كانت أى من هذه الدوال لها خلاصة سالبة سوف تتجاهل الإشارة السالبة وسوف تنفذ الحسابات على أساس القيمة المطلقة للخلاصة . ومع ذلك فداًماً تطبع رسالة خطأ عند حساب قيمة الدالة ، تشير إلى أن قيمة الخلاصة التي أعطيت للدالة كانت قيمة سالبة .

تتطلب دالة INT بعض الإيضاحات الإضافية . تسبب هذه الدالة بتر الخلاصة إذا كانت القيمة الموجبة (أى سوف نسقط من الاعتبار الجزء العشري) . وبذلك سوف تولد دالة INT رقماً صحيحاً موجباً قيمته أقل من خلاصته . ومن الناحية الأخرى إذا كانت الخلاصة لها قيمة سالبة فإن دالة INT سوف تنتج رقماً صحيحاً سالباً قيمته أكبر من خلاصته . وذلك موضعاً في المثال ٥ - ٢ التالي :

مثال ٥ - ٢

فلندرس الجملة .

```
10 LET Y=INT(X)
```

إذا كانت X (الخلاصة) تمثل الرقم 12.9 ، فإن القيمة التي تحدد للمتغير Y هي 12 . ومن الناحية الأخرى ، إذا كانت X تمثل 4.2 - فإن القيمة التي تحدد بها Y هي 5 - .

تمكن دالة TAB المبرمج من توصيف المكان المحدد لكل بند من قائمة بنود المخرجات في جملة PRINT . وذلك يسمح بمرونة أكثر في المباشرة بين بيانات المخرجات من الطرق الموصوفة في قسم ٢ - ١٠ . كل مرة تظهر فيها دالة TAB في قائمة مخرجات ، سوف تتحرك رأس الطباعة أو مؤشر النهاية الطرفية المركزية ناحية اليمين حتى تصل إلى العمود الموصوف (لاحظ أن العمود الموجود أقصى اليسار يعتبر العمود رقم 0) . إذا كان موضع رأس الطباعة فعلاً وراء نطاق (أى إلى يمين) العمود المشار إليه ، فسوف تهمل دالة TAB .

مثال ٥ - ٣

نفرض أننا نريد طباعة القيم A و B و C على سطر واحد ، حيث تطبع القيمة الأولى ابتداء من العمود رقم 9 ، والقيمة الثانية في العمود 29 والقيمة الثالثة في العمود 47 . (لاحظ أن هذه الأعمدة حقيقة هي الأعمدة رقم 10th و 30th و 48th على الترتيب). ويتم ذلك بكتابة :

```
100 PRINT TAB(9);A;TAB(29);B;TAB(47);C
```

ومن الناحية الأخرى إدرس الجملة التالية :

```
150 PRINT "NAME AND ADDRESS";TAB(12);N$
```

سوف يتم تجاهل دالة TAB في هذه الحالة ، حيث رأس الطباعة (أو مؤشر الشاشة) تحركت إلى العمود رقم 16 بعد الانتهاء من طباعة سلسلة الحروف NAME AND ADDRESS . وعلى ذلك فإن سلسلة الحروف الممتلئة بالمتغير N\$ سوف تبدأ في العمود 16 . ومع ذلك ، فإذا كانت خلاصة دالة TAB أكبر من 16 فيمكن تحريك رأس الطباعة (أو مؤشر الشاشة) إلى المكان المطلوب .

استخدام الدالة المكتبية ليس مقصوراً على جملة LET أو جملة PRINT ولكن يمكن ظهور الدالة المكتبية أيها يوجد متغير عادي . وعلاوة على ذلك ، فلا يستلزم أن تكون الخلاصات متغيرات بسيطة بل يمكن استخدام الثوابت والمتغيرات ذات الأدلة والصيغ الرياضية حتى أنه يمكن الرجوع إلى دوال أخرى كخلاصات صحيحة للدالة . وسوف تؤدي عملية البتر أو توماتيكيا عند الضرورة (للدوال التي تتطلب خلاصات ذات قيمة صحيحة (مثلاً كدالة TAB)) .

مثال ٥ - ٤

كل من الجمل التالية أمثلة صحيحة لاستخدام دالة مكتبية :

```
40 LET X1=(-B+SQR(D))/(2*A)
60 IF ABS(X-X1)<.00001 THEN 110
100 PRINT SIN(T),COS(T),TAN(T),LOG(T),EXP(T)
200 FOR J=0 TO INT(Y)
75 LET A=LOG(SQR(ABS(P)))
```

استخدام الدوال المكتبية مشروح في المثال التالى . (لاحظ أن الدوال المكتبية تم استخدامها في الأشكال ٤ - ٢ و ٤ - ١٠) .

مثال ٥ - ٥ جدول دوال A Table of Functions

نفرض أننا نريد تكوين جدول للقيم $\sin(x)$ و $\cos(x)$ و $\tan(x)$ و $\log(x)$ و x^m لعدد 21 قيمة للمتغير x التي تتراوح قيمته ما بين 0 و π . يمكن إنجاز ذلك بسهولة باستخدام حلقة FOR-TO مع البيانات المحسوبة ثم يتم عرضها في شكل عمودى .

المخطط التمهيدى للبرنامج The Program Outline

سوف نجرى مواصلة الحسابات كما يلى :

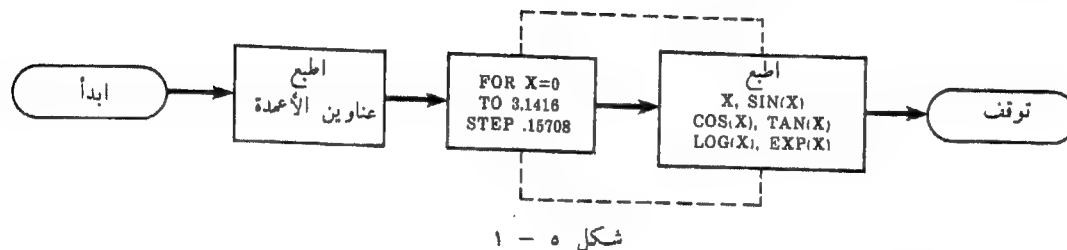
١ - اطبع عناوين الأعمدة الستة التالية :

```
X SIN(X) COS(X) TAN(X) LOG(X) EXP(X)
```

٢ - اطبع القيم x و $\sin(x)$ و $\cos(x)$ و $\tan(x)$ و $\log(x)$ و x^m لكل قيمة من قيم x تتراوح قيمها من $x=0$ إلى $x=3.1416$ بحجم خطوة 0.15708 (والتي تساوى $20 \div 3.1416$). وسوف تولد ذلك 21 سطرًا من المخرجات .

٣ - توقف .

خريطة سير العمليات المناظرة لذلك موضحة في شكل ٥ - ١ .



برنامج البيسك The BASIC Program

مبين في شكل ٥ - ٢ برنامج بيسك كامل، متبوع بالبيانات المحسوبة المخرجة . لا يتطلب البرنامج أي شرح خاص ولكنه برنامج مباشر . ومع ذلك ، فيجب أن نشير إلى توخي الحرص عند تحديد خلاصات الدالة TAB حتى يتسنى طباعة أعمدة المخرجات الستة متباعدة بالتساوي على عرض الصفحة . يجب أن ينفذ البرنامج عدة مرات ، بتعديلات متتالية لخلاصات دالة TAB بعد كل مرة تنفذ فيها ، حتى يتم التوصل إلى تباعد الأعمدة بصورة صحيحة .

تبين مخرجات البيانات بوضوح القيم المحسوبة $\sin(x)$ و $\cos(x)$ و $\tan(x)$ و $\log(x)$ و e^x لكل قيمة من قيم x . لاحظ صدور رسالة خطأ عند محاولة حساب لوغاريتم (\log) للقيمة صفر . ويتبع ذلك الرقم $1.70141E + 38$ - وهو أكبر رقم (في القيمة) سالب يمكن أن يتلاءم مع سعة الحاسب . (وحقيقة فإن لوغاريتم صفر يعرف بأن له قيمة سالبة لا نهائية) . كان يمكن تجنب مثل هذا الموقف بتضمين جملة IF في البرنامج .

```
10 REM GENERATION OF A TABLE OF MATHEMATICAL FUNCTIONS
20 PRINT TAB(7); "X"; TAB(13); "SIN(X)"; TAB(27); "COS(X)";
30 PRINT TAB(39); "TAN(X)"; TAB(51); "LOG(X)"; TAB(63); "EXP(X)"
40 PRINT
50 FOR X=0 TO 3.1416 STEP .15708
60 PRINT TAB(3); X; TAB(13); SIN(X); TAB(25); COS(X);
70 PRINT TAB(37); TAN(X); TAB(49); LOG(X); TAB(61); EXP(X)
80 NEXT X
90 END
```

>RUN

X	SIN(X)	COS(X)	TAN(X)	LOG(X)	EXP(X)
0	0	1	0		1
% LOG OF ZERO IN LINE 70					
-1.70141E+38					
0.15708	0.156435	0.987688	0.158385	-1.851	1.17009
0.31416	0.309018	0.951056	0.324921	-1.15785	1.36911
0.47124	0.453991	0.891006	0.509527	-0.752388	1.60198
0.62832	0.587786	0.809016	0.726545	-0.464706	1.87446
0.7854	0.707108	0.707103	1.	-0.241562	2.19328
0.94248	0.809018	0.587783	1.37639	-5.92406E-2	2.56634
1.09956	0.891008	0.453988	1.96262	9.49101E-2	3.00284
1.25664	0.951057	0.309014	3.07771	0.228441	3.5136
1.41372	0.987689	0.156431	6.31389	0.346225	4.11122
1.5708	1	-3.65144E-6	-273864.	0.451585	4.81049
1.72788	0.987688	-0.156438	-6.31359	0.546895	5.62871
1.88496	0.951055	-0.309021	-3.07764	0.633907	6.58609
2.04204	0.891004	-0.453995	-1.96259	0.713949	7.70631
2.19912	0.809014	-0.587789	-1.37637	0.788057	9.01707
2.3562	0.707103	-0.707111	-0.999989	0.85705	10.5508
2.51328	0.587781	-0.80902	-0.726534	0.921389	12.3454
2.67036	0.453985	-0.891009	-0.509518	0.982213	14.4452
2.82744	0.309011	-0.951059	-0.324912	1.03937	16.9021
2.98452	0.156428	-0.987689	-0.158377	1.09344	19.777
3.1416	-7.25607E-6	-1	7.25607E-6	1.14473	23.1409

TIME: 0.18 SECS.

شكل ٥ - ٢

في الواقع تتضمن جميع نسخ البيسك الدوال المكتبية المبينة في الجدول ٥ - ١ وفي الملحق B . وتتضمن كثير من نسخ اللغة دوالا مكتبية إضافية ، قد يكون بعضها فريداً لهذه النسخة بالذات . ومعظمها لها طبيعة عددية لكن بعضها تقبل خلاصات حرفية أو تعطى نتائج حرفية (أو كليهما) وهذا صحيح وخصوصاً في نسخ بييسك الحاسبات الدقيقة (انظر قسم ٩-٣) . ويجب أن يرجع القارئ إلى المرجع الخاص بجهاز معين ليحدد تماماً الدوال المكتبية المتاحة .

٥ - ٢ القوائم والجداول (المجموعات المتراسة) LISTS AND TABLES (ARRAYS)

عند كتابة برنامج كامل فغالباً من الأفضل أن نشير لمجموعة كاملة من البنود في وقت واحد . مثل هذه المجموعة غالباً ما يشار إليها كمجموعة متراسة . فثلاً ، يمكننا أن نختص بقائمة كاملة من البنود (تعرف أيضاً كمجموعة متراسة ذات بعد واحد) ، أو بكل المداخل في جدول (مجموعة متراسة ذات بعدين) . يسمح لنا البيسك بالإشارة لعناصر من القوائم والجداول كما لو كانت متغيرات طبيعية ، وبذلك يجعل تداول المجموعات المتراسة بسيطاً بقدر الإمكان .

يمكن للعناصر الموجودة في قائمة أو جدول أن تكون إما كيات عددية أو سلاسل حرفية . (تسمح بعض نسخ بييسك بقوائم من السلاسل الحرفية ولكن لا تسمح بجداول من السلاسل الحرفية) . ومع ذلك فيجب أن تكون كل عناصر المجموعة المتراسة من نفس النوع (أى كلها عددية أو كلها سلاسل حرفية) . يجب أن تسمى المجموعة المتراسة التي تحتوى على عناصر رقمية بحرف واحد ، بينما يشار إلى المجموعة المتراسة من السلاسل الحرفية بحرف تتبعه علامة \$ (الدولار) . لاحظ أن أسماء المجموعات المتراسة المكونة من حرف يتبعه رقم أو حرف ثم رقم ثم علامة الدولار ، غير مسموح بها في البيسك التقليدى .

يجب أن يكون إسم المجموعة المتراسة فريداً بداخل البرنامج ، أى لا يمكن لمجموعتين متراسيتين أن تحملتا نفس الإسم . ولكن يمكن لمجموعة متراسة ومتغير عادى أن يكون لهما نفس الإسم . ولكن مثل هذا التكرار في أسماء البرنامج يمكن أن يسبب خلطاً ، وبذلك لا نوصى به .

مثال ٥ - ٦

يحتوى برنامج على قائمة من الأسماء و جدول من الأرقام . سوف يطلق على القائمة \$L وعلى الجدول T .

يمكن أيضاً أن يتضمن البرنامج متغيراً حرفياً عادياً يسمى \$L ومتغيراً رقمياً عادياً يسمى T . سوف تكون هذه المتغيرات منفصلة ومختلفة عن المجموعات المتراسة \$L و T . ولكن من المفضل أن تسمى هذه المتغيرات بأسماء مختلفة (مثال \$L و T9) ، بذلك نتجنب أى خلط ممكن حدوثه بين المجموعات المتراسة وأى متغيرات أخرى في البرنامج .

٥ - ٣ المتغيرات ذات الأدلة SUBSCRIPTED VARIABLES

تعرف العناصر الفردية بداخل مجموعة متراسة كمتغيرات ذات أدلة . ويمكن الرجوع إلى مثل هذا العنصر بذكر إسم المجموعة المتراسة تتبعه قيمة الدليل . وفي حالة الجدول فيجب توصيف دليلين منفصلين بفصلة (و) . وبذلك فإن P(3) عنصر من القائمة P . و T(5,5) عنصر من الجدول T . يجب أن تكون للأدلة قيما صحيحة موجبة ولا يمكن أن تكون قيما سالبة .

مثال ٥ - ٧

المراد وضع عشرة أسماء (سلاسل) في قائمة فردية تسمى \$L ، وسوف يشار للأسماء كما يلي :

\$L(1) و \$L(2) \$L(10) . لاحظ أن الدليل يأخذ أرقاماً صحيحة تتراوح ما بين 1 إلى 10 .

وبالمثل فالمراد تنظيم أربعين رقماً في جدول باسم T له 5 صفوف و 8 أعمدة يمثل أول دليل (رقم الصف) وسوف يأخذ قياً صحيحة تتراوح ما بين 1 إلى 5 ، يمثل الدليل الثاني (رقم العمود) وسوف يتراوح من 1 إلى 8 . وبذلك فإن الرقم الموجود في الصف الثالث والعمود الرابع سوف يشار إليه (3, 4) T . . . الخ .

ليس من الضروري كتابة الدليل ككتاب . ولكن يمكن استخدام المتغيرات والصيغ الرياضية وأسماء الدوال أيضاً . ومع ذلك ، يجب أن تكون قيمة الدليل إما صفراً أو رقماً صحيحاً موجباً . أما إذا كان ناتج الصيغة الرياضية أو مرجع الدالة قيمة غير صحيحة للدليل ، فإن هذه القيمة سوف تبتز ، وبذلك تكون نتيجة للدليل قيمة صحيحة موجبة . أما إذا كانت النتيجة المولدة للدليل قيمة سالبة ، أو رقماً موجباً كبيراً جداً فسيتوقف التنفيذ وتطبع رسالة خطأ .

مثال ٥ - ٨ .

كل المتغيرات ذات الأدلة المبينة فيها بعد مكتوبة بصورة صحيحة :

P(3)	T(8,5)
P(K)	T(J1,J2)
P(C(J))	T(6,N)
P(2*A-B)	T(A1+B1, A2+B2)
P(SQR(X+2+Y+2))	T(ABS(X+Y), ABS(X-Y))

إدرس المتغيرات ذات الدليل P(2*A-B) نفرض أن الصيغة 2*A-B لها قيمة 2.8 ، فإن هذه القيمة سوف تبتز وسوف يفسر المتغير ذو الدليل P(2) أما إذا كانت هذه الصيغة لها القيمة 3.6 - فسوف تطبع رسالة خطأ ويتوقف تنفيذ البرنامج .

يمكن استخدام المتغيرات ذات الأدلة بداخل البرنامج بنفس الطريقة التي تستخدم بها المتغيرات العادية . والمثال التالي يوضح ذلك .

مثال ٥ - ٩ كلمة غير مرتبة Word Unscrambling

مسألة مشوقة تتضمن مداولة المتغيرات ذات الأدلة ألا وهي إعادة تنظيم مجموعة من الحروف لتكون كل الكلمات الممكنة . فلنفرض مثلاً ، أنه تم إعطاؤنا أربعة حروف مثل OPST . ونود تكوين كل التوافقات الممكنة من هذه الحروف الأربعة ثم بعد ذلك الكلمات الممكنة . وبذلك يمكن إيجاد كل الكلمات ذات الحروف الأربعة التي يمكن تكوينها من الحروف الأربعة الأصلية (POST و POTS و TOPS و STOP و SPOT)

طريقة إجراء الحسابات Computational Procedure

ومن أجل عمل ذلك ، دعنا نضع الحروف الأربعة المعطاة في قائمة ونطلق عليها L\$. وسوف يمثل كل حرف بمتغير ذي دليل L\$(1) و L\$(2) و L\$(3) . وبذا يكون الهدف هو كتابة كل التوافقات الممكنة من المتغيرات ذات الدليل .

دعنا نطبع الحروف الأربعة المعطاة بالترتيب الذي يرمز له بالمؤشرات I1 و I2 و I3 و I4 حيث I1 هو دليل أول حرف يجب طباعته ، ويشير I2 للحرف الثاني المطلوب طباعته ، وهكذا . فمثلاً إذا كان I1 = 3 و I2 = 2 و I3 = 4 و I4 = 1 ، فسوف تطبع الحروف بالترتيب .

L\$(3) L\$(2) L\$(4) L\$(1)

ونود أن نكتب برنامج ببسك يسمح أن تأخذ I1 و I2 و I3 و I4 كل القيم الممكنة مع التقييد بأن يأخذ كل مؤشر قيمة عددية فريدة . معنى ذلك أنه لا يسمح لأي مؤشرين أن يكون لهما نفس الرقم .

بالإضافة لجملي INPUT و PRINT ، فإن برنامج البيسك سوف يتكون أساساً من ثلاث حلقات تكرارية متداخلة FOR-TO. أقصى حلقة خارجية تحدد قيمة I1 ، والحلقة التالية تحدد قيمة I2 ، مع عمل اختبار للتأكد أن I2 تختلف عن I1 وسوف تملي الحلقة الداخلية قيماً للمتغير I3 ، ومرة أخرى يجرى اختبار لترى أن I3 لها قيمة مختلفة عن I1 أو I2 وأخيراً ، يمكننا الحصول على قيمة I4 . ملاحظة أن مجموع الأعداد الأربعة الأولى هي 10 . وبذلك فإذا أعطيت I1 و I2 و I3 قيماً منفردة ، يمكننا حساب I4 من الصيغة .

$$I4 = 10 - (I1 + I2 + I3)$$

المخطط التمهيدي للبرنامج The Program Outline

يمكننا عمل مخطط تمهيدي للإجراء كاملاً كما يلي :

١ - اقرأ L\$(1) و L\$(2) و L\$(3) و L\$(4) ،

٢ - نفذ التالى لقسم $I1 = 1, 2, 3, 4$.

(أ) حدد قيمة للمتغير I1

(ب) نفذ التالى مع جعل $I2 = 1, 2, 3, 4$

(i) حدد قيمة للمتغير I2 .

(ii) أجز اختباراً لترى أن I2 تختلف عن I1 ، إذا لم يتحقق ذلك فأضف مقدار الوحدة إلى I2 وأعد الاختبار مرة أخرى .

(iii) لكل قيمة من I1 و I2 اعمل الآتى مع جعل $I3 = 1, 2, 3, 4$

(١) حدد قيمة I3

(٢) أجز اختباراً لترى أن I3 تختلف عن كل من I1 و I2 . إذا لم يتحقق ذلك أضف مقدار الوحدة إلى I3 . وأعد الاختبار مرة أخرى .

(٣) احسب قيمة I4 من الصيغة $I4 = 10 - (I1 + I2 + I3)$

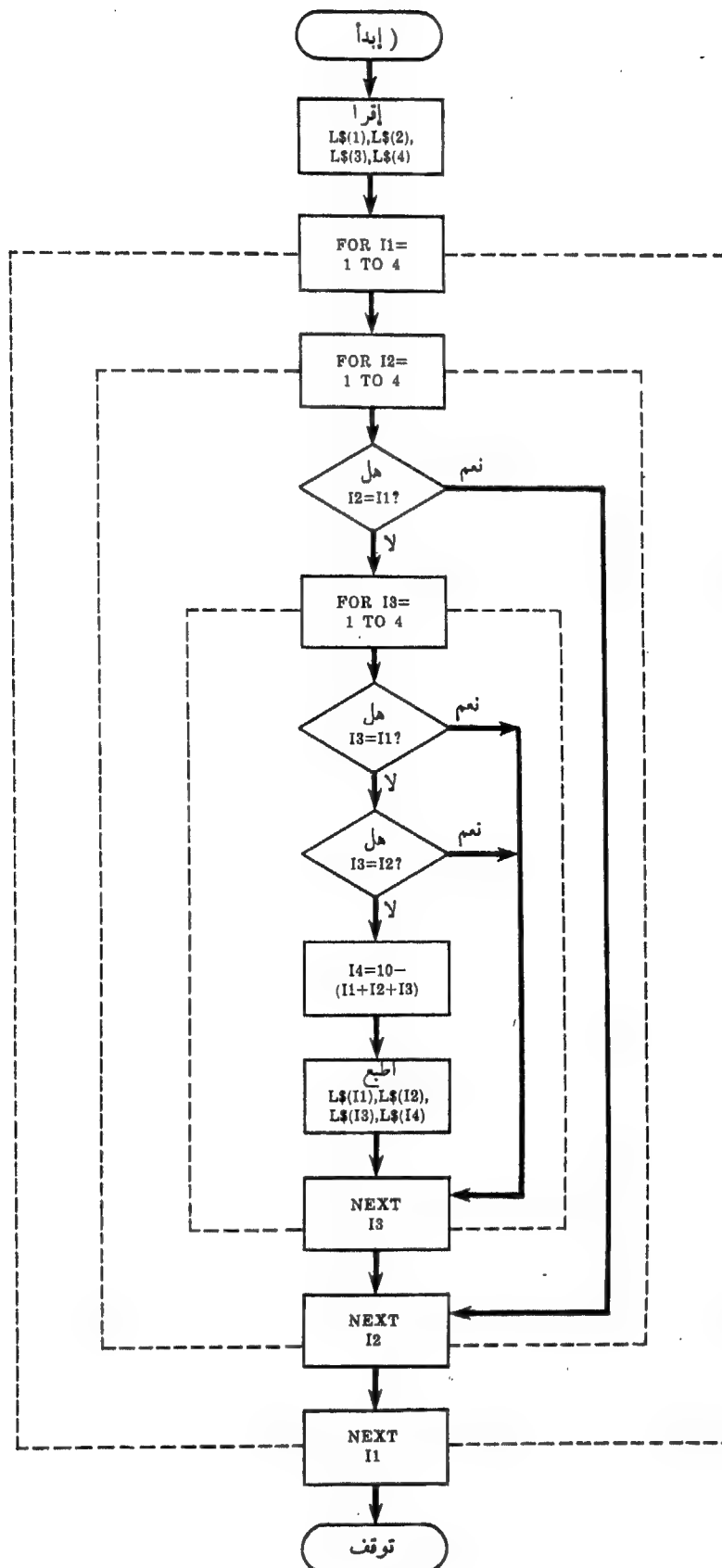
(٤) اطبع L\$(I1) و L\$(I2) و L\$(I3) و L\$(I4) ،

٣ - توقف .

خريطة سير العمليات لهذا الإجراء مبينة في شكل ٥ - ٣ .

برنامج البيسك The BASIC Program

يظهر برنامج البيسك الحقيقى في شكل ٥ - ٤ . ومبين تحت البرنامج المخرجات المولدة للحروف الأربعة OPST . نرى أن هناك 24 طريقة مختلفة يمكن بها توافق الحروف الأربعة . ويمكن برهنة ذلك رياضياً بأن ذلك هو العدد الصحيح للتوافقيات . وبفحص مرئى يمكن إيجاد الكلمات الخمس التى يمكن التعرف عليها وهى : POST و POTS و SPOT و STOP و TOPS . وقد تم إحاطة هذه الكلمات بدوائر في شكل ٥ - ٤ .



شكل ٥ - ٣

```

10 REM FOUR-LETTER WORD UNSCRAMBLER
20 PRINT "TYPE ANY FOUR LETTERS:"
30 PRINT
40 INPUT L$(1),L$(2),L$(3),L$(4)
50 PRINT
60 FOR I1=1 TO 4
70   FOR I2=1 TO 4
80     IF I2=I1 THEN 150
90     FOR I3=1 TO 4
100      IF I3=I1 THEN 140
110      IF I3=I2 THEN 140
120      LET I4=10-(I1+I2+I3)
130      PRINT L$(I1),L$(I2),L$(I3),L$(I4)
140    NEXT I3
150  NEXT I2
160 NEXT I1
170 END

```

> RUN

TYPE ANY FOUR LETTERS:

T O P S T

OPST
 OPTS
 OSPT
 OSTP
 OTPS
 OTSP
POST
POTS
 PSOT
 PSTO
 PTOS
 PTSO
 SOPT
 SOTP
SPOT
 SPTO
STOP
 STPO
TOPS
 TOSP
 TPOS
 TPSO
 TSOP
 TSPO

شكل ٥ - ٤

٥ - ٤ تعريف المجموعات المتراسة — جملة DIM :

DEFINING ARRAYS — THE DIM STATEMENT

يحدد البيسك 11 عنصراً لكل قائمة و 121 عنصراً (11 صفاً و 11 عموداً) لكل جدول يظهر في البرنامج أوتوماتيكياً . وبذلك يسمح لكل دليل أن يتراوح ما بين صفر إلى 10 . وبالطبع قد لا يتطلب الأمر الاستفادة من كل العناصر الموجودة في المجموعة المتراسة — فيمكن بدء الدليل بقيمة صحيحة أكبر من الصفر أنهاء بقيمة صحيحة أقل من 10 . (في مثال ٥-٩ ، مثلا ، يتراوح الدليل من 1 إلى 4 فقط) . ويمكننا أيضاً الاستفادة من مجموعات متراسة أكبر من ذلك . ولعمل ذلك يجب تعريف حجم كل مجموعة متراسة ، أى ، يجب توصيف أقصى عدد من العناصر في كل مجموعة متراسة . ويمكن إنجاز ذلك بواسطة جملة DIM (أبعاد DIMENSION) .

تتكون جملة DIM من رقم للجملة ، تتبعها الكلمة الدالة DIM يتبعها اسم مجموعة متراسة أو أكثر يفصل بينها فصلة (,) . يجب أن يتبع كل اسم لمجموعة متراسة رقم ثابت صحيح أو رقان (رقم واحد للقائمة ورقان للجدول) محصورة بين قوسين ، وعند استخدام رقمين صحيحين يجب أن تفصل بينهما بفصلة (,) . وتشير هذه الأرقام لأقصى قيمة مسموح بها للدليل في المجموعة المتراسة

مثال ٥ - ١٠

يحتوي برنامج على جدول يسمى A وقائمتين رقميتين B و C وقائمة سلسلة حرفية تسمى F\$. مطلوب أن يكون للجدول 50 صفًا و 100 عمودًا وسوف تحتوي القائمتان B و C على 100 عنصر و 50 عنصراً على الترتيب وسوف تحتوي F\$ على 65 عنصراً فيمكن للبرنامج في هذه الحالة أن يحتوي على جملة DIM التالية :

20 DIM A(50,100), B(100), C(50), F\$(65)

تحتجز هذه الجملة حقيقة 51 صفًا و 101 عمودًا للجدول A و 101 عنصراً للقائمة B و 51 عنصراً للقائمة C و 66 عنصراً للقائمة F\$. وبذلك فإن المجموعات المتراسة تكون أكبر قليلاً من المطلوب . ولكن كثيراً من المبرمجين يفضلون أن يبدأ الدليل بالقيمة 1 وليس بالقيمة صفر ، كما لو كانوا يتعاملون مع متغيرات جبرية لها أدلة . وتحت هذه الظروف يكون من الأسهل كتابة جملة DIM بالصورة المبينة عالياً .

يمكن أن تظهر جملة DIM في أى مكان في برنامج البيسك ولكن من الأنفصل عملياً أن تضع جملة DIM عند بداية البرنامج حتى يكون وجودها واضحاً . ويسمح ذلك للمبرمج أو المستفيد أن يحدد أقصى أحجام للمجموعات المتراسة بسهولة وبسرعة .

يمكن أن تتضمن جملة DIM القوائم التي تحتوي على أقل من 11 عنصراً أو الجداول التي تحتوي على أقل من 121 عنصراً ، بالرغم من عدم أهمية عمل ذلك . ويترتب على ذلك حجز كلمات أقل في ذاكرة الحاسب إلا أنه يتطلب بعض الحرص عند التعامل مع الجداول ، لأن أحد الأدلة يمكن أن يتجاوز القيمة 10 حتى لو كان العدد الإجمالي للعناصر أقل من 121 وعند حدوث ذلك يجب استخدام جملة DIM وسوف نرى مثل هذا الموقف في المثال التالي .

مثال ٥ - ١١

برنامج يحتوي على جملة DIM التالية :

30 DIM P(6), Q(10), R(5,15)

ويترتب على ذلك حجز 7 كلمات من المخزن للعناصر الستة للقائمة P و 11 كلمة للقائمة Q و 96 كلمة (6 صفوف 16 عموداً) للجدول R . إتمام P و Q في جملة DIM كان غير مهم حقيقة ، حيث يتم تحديد عدد كاف من المخازن أو توماتيكياً . ومن الناحية الأخرى ، كان يجب أن نضمن الجدول R في جملة DIM حيث أن الدليل الثاني يتعدى 10 وهذا صحيح بالرغم من أن العدد الإجمالي للكلمات المطلوبة بواسطة R أقل من 121 عنصراً .

وسوف نرى مثالا كاملاً لبرنامج يتطلب وجود جملة DIM في جزء لاحق من هذا الفصل .

٥ - ٥ ادخال بيانات الإدخال - جملتي READ و DATA

ENTERING INPUT DATA—THE READ AND DATA STATEMENTS

تتطلب عدة برامج بيسك أن تدخل للحاسب كمية كبيرة من عناصر البيانات ، ويمكن إنجاز ذلك بجملة INPUT ، على الرغم من أن تنفيذ ذلك أمر مرهق . وعادة يكون أكثر ملاءمة إدخال مثل هذه البيانات بواسطة جملتي READ و DATA . تستخدم هاتان الجملتان أيضاً لإدخال البيانات للبرنامج الذي يستخدم أسلوب التشغيل بالدفعات وليس بأسلوب التشغيل بالمشاركة الزمنية .

توصف جملة READ المتغيرات التي يجب أن تدخل قيمها للحاسب . وتتكون هذه الجملة من رقم جملة ، تتبعها الكلمة الدالة READ ، تتبعها قائمة بمتغيرات الإدخال . يمكن أن تحتوي الجملة على كل من المتغيرات العادية والمتغيرات ذات الأدلة مثل قوائم عديدة (/ و) قوائم حرفية . إذا احتوت الجملة على متغيرين أو أكثر فيجب أن تفصل بينهما بفصلة (,) .

والفرض من جملة DATA هو تحديد قيم مناسبة للمتغيرات التي سبق وأن ظهرت في جملة READ . وتتكون جملة DATA من رقم جملة تتبعها الكلمة الدالة DATA ، تتبعها مجموعة من الأرقام (أو / و) السلاسل الحرفية ، تفصل بينها الفاصلة (،) . كل رقم (أو / و) سلسلة حرفية في جملة DATA يجب أن يناظر متغيراً من نفس النوع في جملة READ .

مثال ٥ - ١٢

يحتوى برنامج ببسك على الجمل التالية :

```
30 READ K,N$,Z(1)
```

```
120 DATA 12,SEVENTEEN,-5
```

يترتب على هذه الجملة أن نعطي المتغير K القيمة 12 والمتغير N\$ السلسلة SEVENTEEN والمتغير Z(1) بالقيمة 5 - .

لا يستلزم أن تناظر جملة DATA لكل جملة READ معينة ، برغم أن مثل هذا التناظر جملة تقابل جملة مسموحاً به في حالة الرغبة في ذلك (كما رأينا في المثال السابق) . ولكن النقطة الهامة هي أن كل جملة DATA الموجودة في البرنامج تتجمع وتكون كتلة من قيم البيانات . ويجب أن يناظر كل عنصر في كتلة البيانات متغيراً في جملة READ . يجب أن يكون هذا التناظر بالنسبة لكل من الترتيب والنوع .

مثال ٥ - ١٣

يحتوى برنامج ببسك على الجمل التالية :

```
40 READ A,B,C
```

```
50 READ P$,Q$
```

```
60 READ F(1),F(2),F(3)
```

```
210 DATA 3,-2,11,AB
```

```
220 DATA CD,-8,0,10
```

يترتب على هذه الجمل أن تحدد قيم A و B و C بالأرقام 3 و 2 - و 11 . وتحدد قيم P\$ و Q\$ بالحروف AB و CD وتحدد قيم المتغيرات ذات الأدلة (1) F و (2) F و (3) F بالأرقام 10 و 0 و 8 - . لاحظ أن جملة READ لا تناظر جملة DATA على أساس جملة ، لكن تناظر المتغيرات في جملة READ العناصر في كتلة البيانات من حيث الترتيب والنوع .

وكان يمكن أيضاً جمع جملتي DATA لتكون جملة واحدة .

```
210 DATA 3,-2,11,AB,CD,-8,0,10
```

وسوف تكون النتائج هي نفسها التي عرضت عالية .

يجب أن نشير إلى أن البيانات التي تم توصيفها خلال جملتي READ و DATA هي جزء من البرنامج ، بعكس البيانات التي تم إدخالها خلال جملة INPUT ، ولذلك فإن البيانات التي تحتويها كتلة البيانات تخزن حيناً يخزن البرنامج ، ويتم تعديل قيمها لمجموعة مناسبة من المتغيرات حين يتم تنفيذ البرنامج . وعلى ذلك تكون هذه البيانات دائمة نسبياً ، ويمكن تغييرها فقط بواسطة تعديل جملة DATA أو أكثر بداخل البرنامج .

يجب ملاحظة القواعد التالية عند وضع بنود البيانات في كتلة البيانات :

١ - يجب أن تناظر بنود البيانات قائمة المتغيرات في جملة READ من حيث الترتيب والنوع . ويجب أن يكون عدد العناصر الموجودة في كتلة البيانات مساوياً لعدد المتغيرات في جملة READ وعناصر البيانات الزائدة سوف يتم تجاهلها .

٢ - يجب أن تفصل عناصر البيانات بداخل جملة DATA بواسطة فصلة (,) ولكن يجب ألا يتبع آخر عنصر من البيانات في جملة DATA فصلة (,) .

٣ - يجب أن تتكون عناصر البيانات من أرقام أو سلاسل حرفية . وغير مسموح بالمتغيرات أو الصيغ الرياضية .

٤ - يجب أن تنحصر السلاسل الحرفية التي تحتوى فصالات (,) أو تبدأ بفراغ بين علامتى اقتباس . ويمكن أن تنحصر السلاسل الحرفية الأخرى بين علامتى الاقتباس إذا أردت ذلك .

يمكن لجل DATA أن تظهر في أى مكان من برنامج البيسك ولكنه ، تقليد جيد أن توضع كل جملة DATA متتالية بقرب نهاية البرنامج . وبذلك يكون مكان وتكوين كتلة البيانات واضحاً للغاية .

يوضح المثال التالى تحديد مجموعة من المتغيرات ذات الأدلة بقيم عديدة وكيفية هذه المتغيرات ذات الأدلة . وقد تم أيضاً شرح استخدام جملة DIM .

مثال ٥ - ١٤ إعادة ترتيب قائمة من الأرقام Reordering a List of Numbers

إدرس المسألة الشهيرة وهي إعادة تنظيم قائمة طولها N من الأرقام إلى توال من الأرقام المتزايدة القيمة . يجب كتابة البرنامج بطريقة تسمح بعدم استخدام مخازن أزيد من اللازم . ولذلك فسوف يحتوى البرنامج على مجموعة متراصة واحدة فقط ، حيث يعاد تنظيم عنصر واحد في المرة الواحدة .

طريقة إجراء الحسابات Computational Procedure

سوف يفحص الإجراء القائمة بأكملها بحثاً عن أصغر رقم ويبدل هذا الرقم بأول رقم في القائمة . بعد ذلك تفحص الأرقام $N - 1$ المتبقية بحثاً عن أصغر هذه الأرقام ويبدل هذا الرقم بالرقم الثانى في القائمة ثم بعد ذلك تفحص الأرقام $N - 2$ بحثاً عن أصغر رقم ، ويبدل هذا الرقم بالرقم الثالث في القائمة ، . . وهكذا ، حتى يتم إعادة تنظيم القائمة بأكملها . يتطلب ذلك عدداً من اللفات خلال القائمة مقداره $N - 1$ بالرغم من أن القائمة تتناقص رقماً في كل بحث متعاقب .

من أجل الحصول على أصغر رقم في كل لفة ، فإننا نقارن كل رقم في القائمة بالرقم الذى بدأنا به (ذى الترتيب I) على التوال . إذا كان الرقم ذو الترتيب I أكبر من الرقم ذى الترتيب J مثلاً ، فإننا نبديل الرقين ، وإلا نترك الرقين في أماكنهما الأصلية . فور إتمام تطبيق هذا الإجراء على القائمة بأكملها ، فإن أول رقم في القائمة سوف يكون أصغر رقم . بعد ذلك نعيد الإجراء بأكمله عدد $(N - 2)$ من المرات ، بعدد إجهال $N - 1$ من المرات ($i = 1, 2, \dots, N - 1$) .

السؤال المتبقى الوحيد هو كيف يتم تبديل الرقم ذى الترتيب I مع الرقم ذى الترتيب J . أولاً « نجنب » الرقم ذى الترتيب I للرجوع إليه مستقبلاً ، وذلك يعنى ، أننا نحتفظ بالقيمة الأصلية للمتغير ذى الدليل I . ثم نحدد قيمة المتغير ذى الدليل J بالمتغير ذى الدليل J . وأخيراً ، فإننا نحدد قيمة المتغير ذى الدليل I بقيمة المتغير ذى الدليل J الأصلية والتي سبق تجنبها سابقاً . وبذا يكون قد تم التبديل المطلوب .

المخطط التمهيدى للبرنامج The Program Outline

يمكن القيام بتنفيذ الحسابات بالكامل بواسطة حلقتين FOR-TO ولنر كيف يمكن إنجاز ذلك ، عليك بدراسة التخطيط التمهيدى التالى :

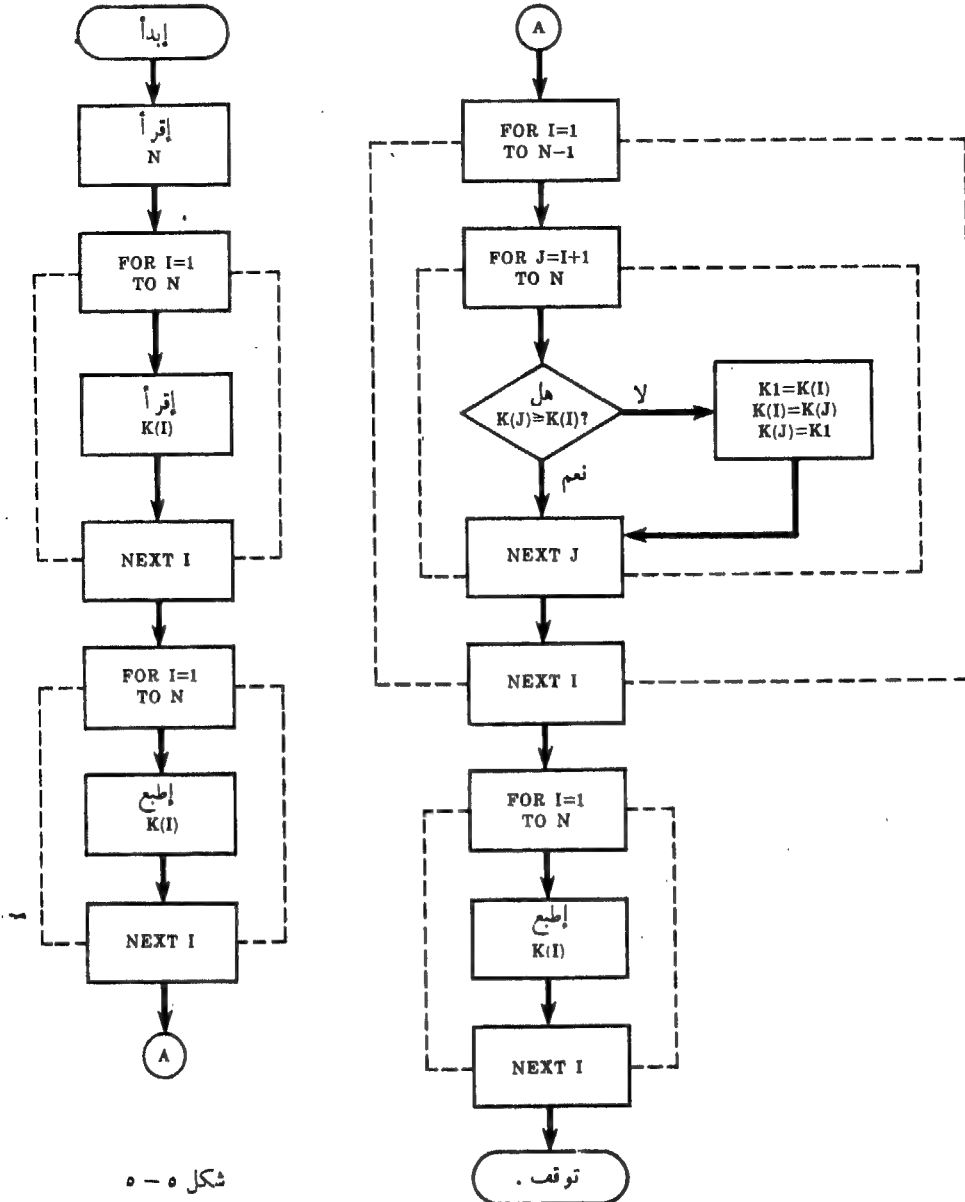
١ - اقرأ حجم القائمة (N) .

٢ - اقرأ ثم اطبع قائمة بها عدد N من الأرقام الصحيحة الثابتة

- ٣ - نفذ الحسابات التالية عدد $(N-1)$ من المرات ، مع جعل $(i = 1, 2, \dots, N-1)$ قارن الرقم ذا الترتيب i وسوف يكون أول رقم في القائمة ، مع كل رقم لاحق [أى كل رقم ذو ترتيب j ، حيث $(j = i+1, i+2, \dots, N)$] وعندما يكون ذو الترتيب i أكبر من الرقم ذو الترتيب j ، نستبدل الرقمين .
- ٤ - إطبّع قائمة الأرقام بعد إعادة تنظيمها .
- ٥ - توقف .
- يوضح الشكل ٥ - ٥ خريطة لسير العمليات لهذا الإجراء .

برنامج البيسك The BASIC Program

عند كتابة برنامج البيسك الفعل دعنا نطلق اسم K على الأرقام الصحيحة وسوف نفترض أن K لن تتكون من أكثر من 101 عنصراً وبذلك فسوف نعمل أقصى قيمة مسوح بها للأدلة هي 100 .



شكل ٥ - ٥

برنامج البيسك موضح في شكل ٥ - ٦ . لاحظ جملة DIM بقرب بداية البرنامج (سطر 20) ، وهي تحدد إن K يمكن أن تحتوي عدداً من العناصر يصل إلى 101 عنصراً . ولاحظ أيضاً استخدام حلقات FOR-TO من أجل قراءة وطباعة عناصر K (مثال ، أنظر السطور 40 إلى 60 و 90 إلى 110 و 330 إلى 350) .

يمكن القيام بإعادة التنظيم بواسطة حلقتين من حلقات FOR-TO كما سبق وصفها (السطور 140 إلى 290) . تضمن البرنامج ثلاث جمل REM . وتساعد هذه الجمل في توضيح منطق الحلقتين FOR-TO .

وأخيراً ، لاحظ وجود جملة DATA قرب نهاية البرنامج (السطور 360 و 370) محتوية على 21 قيمة . أولى هذه القيم (الرقم 20) هي قيمة N ، تشير إلى أن K سيكون بها 20 عنصراً في هذا المثال . وتتبع هذا الرقم القيم العشرون الموجودة في K ، بترتيبها الطبيعي (أى $K(1) = 595$ و $K(2) = 78$ و $K(3) = 1505$ و $K(20) = 710$) .

تظهر المخرجات التي يولدها هذا المثال عند نهاية شكل ٥ - ٦ . نرى القائمة الأصلية للأرقام في أول كشف المخرجات ، وتتبعها فوراً قائمة بالأرقام التي تم إعادة تنظيمها بعد ذلك .

```

10 REM PROGRAM TO REARRANGE A LIST OF NUMBERS INTO ASCENDING ORDER
20 DIM K(100)
30 READ N
40 FOR I=1 TO N
50   READ K(I)
60 NEXT I
70 PRINT "ORIGINAL LIST OF NUMBERS:"
80 PRINT
90 FOR I=1 TO N
100   PRINT K(I);
110 NEXT I
120 PRINT
130
140 REM          REPEAT INTERCHANGE N-1 TIMES
150
160 FOR I=1 TO N-1
170
180   REM          FIND SMALLEST NUMBER IN LIST AND INTERCHANGE
190
200   FOR J=I+1 TO N
210     IF K(J)<K(I) THEN 280
220
230   REM          INTERCHANGE K(J) AND K(I)
240
250   LET K1=K(I)
260   LET K(I)=K(J)
270   LET K(J)=K1
280 NEXT J
290 NEXT I
300 PRINT
310 PRINT "REORDERED LIST OF NUMBERS:"
320 PRINT
330 FOR I=1 TO N
340   PRINT K(I);
350 NEXT I
360 DATA 20,595,78,1505,891,29,7,18,191,36,68,7051,509,212,44,726,1806
370 DATA 289,401,1488,710
380 END

```

>RUN

```

EXS.14          09:32          15-MAR
ORIGINAL LIST OF NUMBERS:
595 78 1505 891 29 7 18 191 36 68 7051 509 212 44 726
1806 289 401 1488 710
REORDERED LIST OF NUMBERS:
7 18 29 36 44 68 78 191 212 289 401 509 595 710 726
891 1488 1505 1806 7051
TIME: 0.09 SECS.

```

تم معالجة عناصر الجدول بطريقة تشبه كثيراً نفس الطريقة التي تم بها معالجة عناصر القائمة . يتطلب دائماً حلقتان من حلقات FOR-TO لعمليات الإدخال / الإخراج التي تجري عند معالجة عناصر الجدول . وسترى توضيحاً لذلك في المثال التالي .

مثال ٥ - ١٥ معالجة عناصر الجدول Table Manipulation

إدرس جدول الأرقام المئين في الجدول ٥ - ٢ . ولنفرض أننا نرغب في تجميع كل العناصر الموجودة في كل صف من الجدول ، كذا كل العناصر الموجودة في كل عمود ، دعنا نكتب برنامج بييسك يسمح لنا بالقيام بهذه الحسابات الأولية .

جدول ٥ - ٢

6	0	-12	4	17	21
-8	15	5	5	-18	0
11	3	1	-17	12	7
13	2	13	-9	24	4
-27	-3	0	14	8	-10

المخطط التمهيدى للبرنامج The Program Outline

سوف نواصل عمل الحسابات طبقاً للتخطيط التالي :

- ١ - إقرأ القيم التي تشير إلى عدد الصفوف M وعدد الأعمدة N .
- ٢ - إقرأ عناصر الجدول (T) على أساس صف بصف بعمل التالي ، لكل قيمة من قيم عداد الصفوف I الذي تتراوح قيمة ما بين 1 إلى M .
 - لكل قيمة I ، اجعل J (عدد الأعمدة) يتراوح ما بين 1 إلى N . إقرأ قيمة للعنصر T(I, J) لكل قيمة من قيم I و J .
- ٣ - اطبع عناصر T على أساس صف بصف . إذن فلكل قيمة I تتراوح ما بين 1 إلى M نتبع مايلي :
 - لكل قيمة من I اجعل J تتراوح ما بين 1 إلى N اطبع قيمة للعنصر T(I, J) لكل قيمة من قيم I و J .
- ٤ - احسب ثم اطبع مجموع عناصر كل صف ، ثم المجموع لمجموع الصفوف (أي المجموع التراكمي لكل الصفوف) كما يلي :
 - (أ) اجعل القيمة المبدئية للمجموع التراكمي (S1) صفراً .
 - (ب) نفذ مايلي بالنسبة لكل قيمة من قيم I التي تتراوح ما بين 1 إلى M :
 - (i) اجعل القيمة المبدئية لمجموع الصف S(I) صفراً .
 - (ii) لكل قيمة من قيم J التي تتراوح ما بين 1 إلى N أضف قيمة T(I, J) إلى قيمة S(I) ، أي ، اجعل :

$$S(J)=S(J)+T(I,J)$$
 - (iii) اطبع القيمة الحالية I والقيمة المناظرة S(I) .
 - (iv) أضف القيمة S(I) للقيمة S1 ، أي ، اجعل :

$$S1=S1+S(J)$$
 - (ح) اطبع القيمة النهائية S1 (أي المجموع التراكمي لكل المجاميع التي سبق حسابها للصفوف) .

هـ - احسب ثم اطبع مجموع العناصر في كل عمود ومجموع مجاميع كل الأعمدة المفردة (أى المجموع التراكمي لكل الأعمدة) كمايلي :

(أ) اجعل القيمة الأولية للمجموع التراكمي S1 مساوياً لصفر .

(ب) اعمل الآتي لكل قيمة من قيم J التي تتراوح من 1 إلى N .

(i) اجعل القيمة المبدئية لمجموع العمود S(J) ، مساوياً لصفر .

(ii) لكل قيمة من I التي تتراوح ما بين 1 إلى M ، أضف قيمة T(I, J) إلى قيمة S(J) أى ، اجعل :

$$S(J) = S(J) + T(I, J)$$

(iii) اطبع القيمة الحالية J والقيمة المناظرة S(J) .

(iv) أضف قيمة S(J) إلى S1 ، أى اجعل :

$$S1 = S1 + S(J)$$

(حـ) اطبع القيمة النهائية S1 (أى ، المجموع التراكمي لمجموع كل الحسابات السابقة لمجاميع الأعمدة) . لاحظ أن المجموع

التراكمي للأعمدة يجب أن يساوى المجموع التراكمي للصفوف ، إذن يجب أن تكون قيمة S1 مساوية للقيمة المطبوعة في الخطوة رقم 4 . ويستخدم ذلك كاختبار .

٦ - توقف .

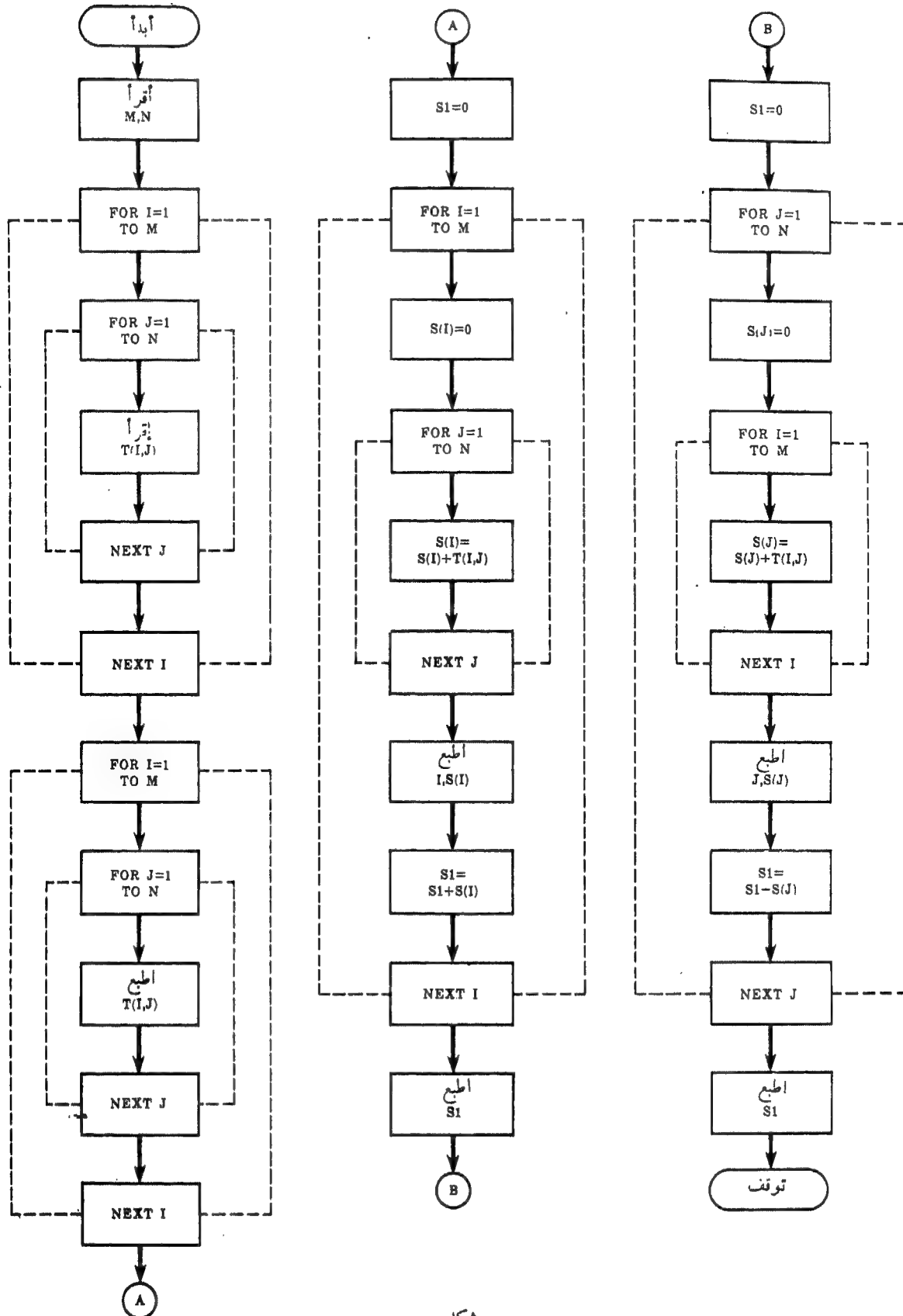
يحتوى شكل ٥ - ٧ على خريطة سير العمليات التي تناظر التخطيط السابق .

برنامج البيسك The BASIC Program

يقدم شكل ٥ - ٨ برنامج بيسك كاملاً لهذه المسألة . لاحظ الحلقات FOR-TO المتداخلة التي تستخدم لقراءة وطباعة عناصر الجدول (أنظر السطور 70 إلى 110 و 170 إلى 220) . استخدمت أيضاً حلقتان FOR-TO لتكوين مجاميع الصفوف (السطور 290 إلى 330) ومجاميع الأعمدة (السطور 460 إلى 540) .

تكون جملتا DATA قرب نهاية البرنامج (السطور 580 و 590) كتلة بيانات بها 32 عنصراً . الرقمان الأولان هما قيمتا M و N ، حيث تشير في هذا المثال إلى أن الجدول يحتوى على 5 صفوف و 6 أعمدة . والأرقام المتبقية وعددها 30 هي قيم عناصر الجدول T ، على أساس صف بصف (وذلك يتطلب زيادة الدليل الثانى J بسرعة عند قراءة البيانات) .

نرى أن البرنامج لا يتضمن DIM حيث لا تعتمد قيم الأدلة 10. إذا تطلب الأمر زيادة حجم المجموعة المتراسة ، تكون جملة DIM ، بالطبع ، مهمة . وأخيراً ، لاحظ أن البرنامج يحتوى على عدد من جمل REM وعدد من الأسطر الخالية ، التي تجعل قراءتها أسهل وتعطى بعض التوضيحات لمنطق البرنامج .



```

10 REM PROGRAM TO SUM ROWS AND COLUMNS OF A TABLE
20
30 READ M,N
40
50 REM READ ELEMENTS OF TABLE
60
70 FOR I=1 TO M
80   FOR J=1 TO N
90     READ T(I,J)
100   NEXT J
110 NEXT I
120
130 REM PRINT TABLE
140
150 PRINT "GIVEN TABLE:"
160 PRINT
170 FOR I=1 TO M
180   FOR J=1 TO N
190     PRINT T(I,J);
200   NEXT J
210   PRINT
220 NEXT I
230 PRINT
240
250 REM SUM ACROSS EACH ROW
260
270 PRINT "SUM OF COLUMNS IN EACH ROW:"
280 PRINT
290 LET S1=0
300 FOR I=1 TO M
310   LET S(I)=0
320   FOR J=1 TO N
330     LET S(I)=S(I)+T(I,J)
340   NEXT J
350   PRINT "ROW ";I,"SUM=";S(I)
360   LET S1=S1+S(I)
370 NEXT I
380 PRINT
390 PRINT "SUM OF ROW SUMS=";S1
400 PRINT
410
420 REM SUM DOWN EACH COLUMN
430
440 PRINT "SUM OF ROWS IN EACH COLUMN:"
450 PRINT
460 LET S1=0
470 FOR J=1 TO N
480   LET S(J)=0
490   FOR I=1 TO M
500     LET S(J)=S(J)+T(I,J)
510   NEXT I
520   PRINT "COLUMN ";J,"SUM=";S(J)
530   LET S1=S1+S(J)
540 NEXT J
550 PRINT
560 PRINT "SUM OF COLUMN SUMS=";S1
570
580 DATA 5,6,4,0,-12,4,17,21,-8,15,5,5,-18,0,11,3,1,-17,12,7
590 DATA 13,2,13,-9,24,4,-27,-3,0,14,8,-10
600 END

```

شكل هـ - ٨

نرى في شكل هـ - ٨ المخرجات التي تم توليدها بواسطة هذا المثال . يطبع الجدول المعطى أولاً ، يتبعه مجموع الصفوف (عناصر) لكل صف والمجموع التراكمي لمجموع الصفوف . ويتبع هذه القيم مجاميع الأعمدة المختلفة والمجموع التراكمي للأعمدة . لاحظ أن المجموع التراكمي للصفوف مساو للمجموع التراكمي للأعمدة (81) ، كما هو متوقع .

GIVEN TABLE:

6	0	-12	4	17	21
-8	15	5	5	-18	0
11	3	1	-17	12	7
13	2	13	-9	24	4
-27	-3	0	14	8	-10

SUM OF COLUMNS IN EACH ROW:

ROW	1	SUM=	36
ROW	2	SUM=	-1
ROW	3	SUM=	17
ROW	4	SUM=	47
ROW	5	SUM=	-18

SUM OF ROW SUMS= 81

SUM OF ROWS IN EACH COLUMN:

COLUMN	1	SUM=	-5
COLUMN	2	SUM=	17
COLUMN	3	SUM=	7
COLUMN	4	SUM=	-3
COLUMN	5	SUM=	43
COLUMN	6	SUM=	22

SUM OF COLUMN SUMS= 81

شكل ٥ - ٩

٥ - ٦ إعادة قراءة البيانات — جملة RESTORE

REREADING DATA—THE RESTORE STATEMENT

رأينا في القسم السابق أنه يجب الاحتفاظ دائماً بالتناظر بين المتغيرات التي تقرأ قيمها (أي قائمة المتغيرات في جملة READ) وعناصر البيانات الفردية في كتلة البيانات (الأرقام والحروف في جملة DATA). ويتم إنجاز ذلك بواسطة « مؤشرات » داخلية حيث تشير إلى العنصر التالي المطلوب قراءته من البيانات . وفي الحقيقة ، يحتفظ بمؤشرين إذا كانت كتلة البيانات تحتوى على بيانات عددية وحرفية — أحد هذين المؤشرين خاص بالأرقام (الثوابت العددية) والآخر للحروف . في كل مرة تتم قراءة عنصر من البيانات ، يتقدم المؤشر أوتوماتيكياً لعنصر البيانات التالي الذي له نفس النوع .

توجد أنواع معينة من المسائل تتطلب قراءة بعض وربما كل العناصر أكثر من مرة . ولعمل ذلك يجب إعادة أحد المؤشرين أو كليهما إلى بداية كتلة البيانات . وتستخدم جملة RESTORE لهذا الغرض .

وتتكون جملة RESTORE من رقم جملة ، تتبعها الكلمة الدالة RESTORE . ويسبب ظهور هذه الجملة إعادة كل من المؤشرين إلى أول عنصر من البيانات من النوع المناظر له بداخل كتلة البيانات .

مثال ٥ - ١٦

يحتوى برنامج بيسك على الجمل التالية :

```

30 READ A,B,C
...
60 RESTORE
70 READ W,X,Y,Z
...
200 DATA 1,3,5,7,9,11,13

```

تسبب الجملة 30 تحديد المتغير A بالقيمة 1 والمتغير B بالقيمة 3 والمتغير C بالقيمة 5 . وعند مصادفة الجملة رقم 60 فسوف يعود المؤشر إلى أول رقم في كتلة البيانات . إذن فالجملة رقم 70 تسبب تحديد المتغير W بالقيمة 1 والمتغير X بالقيمة 3 والمتغير Y بالقيمة 5 والمتغير Z بالقيمة 7 . إذا كانت جملة RESTORE غير موجودة فإن W سوف تحدد بالقيمة 7 و X بالقيمة 9 ، ... وهكذا .

لاحظ أننا اهتمنا بمؤشر واحد فقط في هذا المثال حيث أن كتلة البيانات تحتوي على ثوابت رقمية فقط .

يُسمح للكلمة الثالثة RESTORE أن تتبعها علامة (*) أو علامة () ، وبعض نسخ البيسك عند وجود (*) فإن مؤشر الأرقام فقط هو الذي يعاد مكانه ، بينما يعاد مؤشر سلاسل الحروف إلى مكانه في حالة وجود (\$) فقط . بينما لا يسمح بوجود كل من (*) ، (\$) معاً في نفس الجملة .

مثال ٥ - ١٧

يحتوي برنامج بيسك على الجمل التالية :

```
50 READ A,B,M$,N$
...
150 RESTORE*
160 READ C1,C2,F1$,F2$
...
250 DATA 2,4,RED,GREEN,6,8,BLUE,WHITE
```

تسبب الجملة 50 تحديد المتغيرين A و B بالقيم 2 و 4 والمتغيرين M\$ و N\$ بالحروف RED و GEERN . أعيد المؤشر الرقمي بواسطة الجملة 150 ومن ثم تسبب الجملة 160 تحديد المتغيرين C1 و C2 بالقيم 2 و 4 ، بينما سيتم تمثيل المتغيرات F1\$ و F2\$ بالحروف BLUE و WHITE .

إذا تغيرت الجملة 150 إلى :

```
150 RESTORE$
```

بذلك يعاد المؤشر الحرفي دون المؤشر الرقمي . ولذلك يتم تحديد C1 و C2 بالقيم 6 و 8 ولكن المتغيرين F1\$ و F2\$ سيتم تحديد قيمهما بالحروف RED و GREEN على الترتيب .

والآن نفرض أن الجملة 150 تغيرت إلى :

```
150 RESTORE
```

بذلك يعاد كل من المؤشرين وتكون نتيجة ذلك : $C1 = 2$ و $C2 = 4$ و $F1\$ = RED$ و $F2\$ = GREEN$. وأخيراً ، إذا ألغيت الجملة رقم 150 بالكامل ، فإن نتيجة الجملة 160 تكون $C1 = 6$ و $C2 = 8$ و $F1\$ = BLUE$ و $F2\$ = WHITE$.

يجب أن يتضح أن القيم التي نحدد بها A و B و M\$ و N\$ لا تتأثر بجمل RESTORE و READ التالية في هذا المثال .

٥ - ٧ ملاحظات ختامية CLOSING REMARKS

إننا بهذا الفصل ننهي مناقشتنا للمصائص « الأساسية » للغة البيسك ولقد رأينا أن استخدام الدوال المكتوبة يمكن المبرمج من القيام بتنفيذ عمليات معينة (مثل بتر الجزء العشري ، تباعد بيانات المخرجات ، ... الخ) وذلك بجانب العمليات الرياضية الشائعة . يمكن تخزين والتعامل مع مجموعات من الأرقام والحروف وذلك باستخدام القوائم والجداول . ويمكن لكميات كبيرة من البيانات أن تخزن بطريقة ملائمة وتعمل للمتغيرات في البرنامج أو لعناصر مجموعة متراسة وذلك من خلال استخدام جملتي READ و DATA . يترتب على هذه المصائص تيسير برمجة كثير من المسائل المتنوعة .

اسئلة للمراجعة

Review Questions

- ١ - ما هو الدوال المكتبية ؟ وما هو الغرض الاساسى من استخدامها ؟
- ٢ - ما هي الاسماء الأخرى التى تطلق أحياناً على الدوال المكتبية ؟
- ٣ - اذكر عدة أسماء من الدوال المكتبية الأكثر شيوعاً ؟ ثم اذكر الغرض من كل من هذه الدوال المكتبية ؟
- ٤ - ما هو المقصود من الخلاصة ؟ وهل تتطلب كل الدوال المكتبية خلاصات ؟
- ٥ - كيف تستخدم الدوال المكتبية فى برنامج بيسك ؟
- ٦ - ماذا يحدث لو أعطيت دالة مكتبية قيمة سالبة ولكنها تتطلب خلاصة موجبة ؟
- ٧ - ما هو المقصود من البتر ؟ اذكر مثالا يوضح البتر .
- ٨ - ما هو الغرض من دالة INT ؟ ماذا يحدث عندما تستقبل دالة INT خلاصة موجبة ؟ وعندما تستقبل خلاصة سالبة ؟
- ٩ - ما هو الغرض من دالة TAB ؟ وفى أى جملة تستعمل ؟
- ١٠ - هل يمكن استخدام صيغة رياضية كخلاصة لدالة مكتبية ؟ وهل يمكن الرجوع لدالة مكتبية أخرى لهذا الغرض ؟
- ١١ - ماذا يحدث لو أعطينا قيمة غير صحيحة كخلاصة لدالة مكتبية مع العلم أنها تتطلب خلاصة صحيحة ؟
- ١٢ - ما هو المقصود من القائمة ؟ الجدول ؟
- ١٣ - ما هو المقصود من مجموعة متراسة ذات بعد واحد ؟ مجموعة متراسة ذات بعدين ؟ قارن إجابتك بإجابة السؤال السابق .
- ١٤ - ما هو المقصود بعناصر القائمة أو الجدول ؟ وماذا تمثل هذه العناصر ؟
- ١٥ - هل يمكن أن تحتوى مجموعة متراسة واحدة أرقاماً وحروفاً ؟
- ١٦ - ما هو المتغير ذا الدليل ؟ كيف نشير إلى متغير ذى دليل معين ؟
- ١٧ - هل يمكن استخدام الصيغة الرياضية كدليل ؟ وهل يمكن الإشارة إلى دالة مكتبية واستخدامها لنفس الغرض ؟
- ١٨ - ما هي القيود التى تنطبق على القيم التى يمكن أن تأخذها الأدلة ؟
- ١٩ - ما هو الغرض من جملة DIM ؟ ومتى يجب أن تظهر هذه الجملة فى برنامج بيسك ؟
- ٢٠ - لخص قواعد كتابة جملة DIM .
- ٢١ - هل يمكن لصيغة رياضية أو مرجع لدالة مكتبية أن تظهر فى جملة DIM ؟
- ٢٢ - هل يوجد غرض ما فى توصيف حجم مجموعة متراسة صغيرة فى جملة DIM ؟ اشرح .
- ٢٣ - ما هو الغرض من جملتي READ و DATA ؟
- ٢٤ - لخص قواعد كتابة جملة READ .
- ٢٥ - لخص قواعد كتابة جملة DATA .
- ٢٦ - هل تتطلب كل جملة READ جملة بيانات خاصة بها ؟ اشرح .
- ٢٧ - ما هو المقصود من كتلة البيانات ؟ وما تكون كتلة البيانات ؟
- ٢٨ - هل البيانات التى تدخل من خلال جملة INPUT دائمة كاليانات التى تدخل من خلال جملتي READ-DATA ؟ اشرح .
- ٢٩ - لخص القواعد التى يجب أن تلاحظ عند وضع عناصر البيانات فى كتلة البيانات .
- ٣٠ - أين توضع جملة DATA غالباً فى برنامج بيسك ؟ ولماذا ؟
- ٣١ - كيف يمكن القيام بعمليات الإدخال / الإخراج لقائمة أو جدول ؟
- ٣٢ - ناقش الغرض من المؤشرات كذا ناقش استخدامها مقترنة بكتلة البيانات .
- ٣٣ - ما هو الغرض من جملة RESTORE ؟ اذكر ثلاث طرق مختلفة يمكن أن تكتب بها هذه الجملة .

مسائل محلولة Solved Problems

٥ - ٣٤ أكتب جملة بيسك تناظر كلا من المعادلات الجبرية التالية :

$$z = \tan t \quad (أ)$$

10 LET Z=TAN(T)

$$w = \log_e(v) \quad (ب)$$

10 LET W=LOG(V)

$$y = ae^{bx} \sin cx \quad (ج)$$

10 LET Y=A*EXP(B*X)*SIN(C*X)

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (د)$$

10 LET X1=(-B+SQR(B^2-4*A*C))/(2*A)

٥ - ٣٥ أكتب جملة بيسك لكل من المواقع التالية :

(أ) قرر القيمة المطلقة للفرق بين المتغيرين U و V ثم حدد قيمة المتغير W بالناتج .

10 LET W=ABS(U-V)

(ب) قرر إشارة x . فإذا كانت x سالبة تفرع للجملة 50 ، وإذا كانت x مساوية للصفر تفرع للجملة 20 ، وإذا كانت x موجبة تفرع للجملة 170 .

100 ON SGN(X)+2 GO TO 50,20,170

(ج) قرر أكبر رقم صحيح لا يتعدى قيمة z جبرياً ، حيث $z = x^2 - y^2$ ثم اعط هذا الرقم الصحيح للمتغير I .

10 LET I=INT(X^2-Y^2)

(د) في (ج) عالية ، إذا كانت $x = 2.5$ و $y = 6.3$ ما هي القيمة التي تعطى للمتغير I ؟

$I = -34$ ومن ثم $x^2 - y^2 = -33.44$

(هـ) اطبع القيم X\$ و X و Y\$ و Y على نفس السطر . اجعل السلسلة الممثلة بالمتغير X\$ تبدأ من العمود رقم 10 ، ويتبعها فوراً قيمة Y . وبالمثل اجعل Y\$ تبدأ من العمود رقم 46 ، ويتبعها فوراً قيمة Y .

100 PRINT TAB(9);X\$;X;TAB(45);Y\$;Y

٥ - ٣٦ يبين كل مثال ما يأتي مرجعاً إلى متغير أو أكثر من المتغيرات ذات الأدلة . صف نوع المجموعة المتراسة التي يشار إليها في كل حالة :

(أ) 10 DIM C1(100),N\$(100,3) (أ)

C1 قائمة عددية ، N\$ جدول حرفي .

(ب) 50 LET P(I)=P(I)+Q(I,J) (ب)

P قائمة عددية ، Q جدول رقمي

100 IF A\$(5)=G\$ THEN 220 (ج)

A\$ قائمة حرفية

150 ON X(K(I),J(I)) GO TO 100,20,180,20,250 (د)

X جدول رقمي و K و J قوائم رقمية .

200 PRINT X\$(K),N1(K),N2(K) (هـ)

X\$ قائمة حرفية ، N1 و N2 قوائم رقمية

٣٧- اكتب جملة أو أكثر للقيام بكل من العمليات الآتية :

(أ) اجمع العناصر الـ 100 الأولى من القائمة الرقمية T .

```
10 LET S=0
20 FOR I=0 TO 99
    LET S=S+T(I)
40 NEXT I
```

(ب) اطبع الأرقام الزوجية من القائمة الرقمية T لقيم الدليل التي تتراوح ما بين 0 و 100 أى اطبع
T(0) و T(2) و T(4) و T(100) .

```
10 FOR I=0 TO 100 STEP 2
20 PRINT T(I);
30 NEXT I
```

(ج) احسب مجموع كل عناصر الجدول الرقمي P . اجعل M تشير إلى عدد الصفوف و N تشير إلى عدد الأعمدة .

```
10 LET S=0
20 FOR I=1 TO M
30 FOR J=1 TO N
40 LET S=S+P(I,J)
50 NEXT J
60 NEXT I
```

(د) اطبع عناصر العمود الثالث من الجدول الحرفي K\$. اعرض المخرجات في صورة عمود ابتداء من العمود رقم 12
(يشير رقم العمود هنا إلى موضع الطباعة على الورق) .

```
10 FOR I=1 TO M
20 PRINT TAB(12);K$(I,3)
30 NEXT I
```

٣٨- اكتب جمل READ و DATA مناسبة لكل موقف موصوف فيما يلي :

(أ) حدد القيم 6 — 1.6 E-500، 4077، MAY، و OCTOBER، 100، 110، 120، 130، 140 و 150

المتغيرات و C1 و C2 و C3 و X\$ و Y\$ و Z(1) و Z(2) و Z(3) و Z(4) و Z(5) و Z(6) . اطبع كل متغير

ذى دليل منفصلاً في جملة READ .

```
10 READ C1,C2,C3,X$,Y$,Z(1),Z(2),Z(3),Z(4),Z(5),Z(6)
```

```
200 DATA -1.6E-6,-500,4077,MAY,OCTOBER,100,110,120,130,140,150
```

يمكن أيضاً تقسيم جمل READ و DATA عند الرغبة ، مثلاً ،

```
10 READ C1,C2,C3,X$,Y$
20 READ Z(1),Z(2),Z(3),Z(4),Z(5),Z(6)
...
200 DATA -1.6E-6,-500,.4077
210 DATA MAY,OCTOBER,100,110
220 DATA 120,130,140,150
```

(ب) حدد القيم المغطاة في الجزء (أ) عالية لتغييراتها بالترتيب . استخدم حلقة FOR-TO لعناصر المجموعة المتراسة .

```
10 READ C1,C2,C3,X$,Y$
20 FOR I=1 TO 6
30   READ Z(I)
40 NEXT I
...
200 DATA -1.6E-6,-500,.4077,MAY,OCTOBER
210 DATA 100,110,120,130,140,150
```

(ج) حدد القيم المغطاة في الجزء (أ) عالية لتغييراتها بالترتيب كما في جزء (ب) . وعند جزء لاحق في البرنامج أعد مؤشر السلاسل الحرفية وحدد القيم الحرفية MAY و OCTOBER المتغيرين F\$ و G\$.

```
10 READ C1,C2,C3,X$,Y$
20 FOR I=1 TO 6
30   READ Z(I)
40 NEXT I
...
100 RESTORE$
110 READ F$,G$
...
200 DATA -1.6E-6,-500,.4077,MAY,OCTOBER
210 DATA 100,110,120,130,140,150
```

مسائل تكميلية

Supplementary Problems

٥ - ٣٩ أوجد الدوال المكتتبية المتاحة على جهازك . (أشـر إلى مرجع البيسك الذي تم نشره بواسطة صانع الحاسب الخاص بك) صف غرض كل دالة مكتتبية وقرر تماماً كيف يرجع إلى كل دالة .

٥ - ٤٠ اكتب جملة بيـسك تناظر كلا من المعادلات الجبرية التالية :

$$y = \sqrt{\sin x - \cos x} \quad (أ)$$

$$p = qe^{-at} \quad (ب)$$

$$c = \log_e \sqrt{|a+b|} + \log_e \sqrt{|a-b|} \quad (ج)$$

$$w = ||u-v| - |u+v|| \quad (د)$$

$$z = \cos(x + \arctan y) \quad (هـ)$$

٥ - ٤١ اكتب جملة بيـسك لكل من المواقف التالية :

(أ) قرر ماهي إشارة الكمية $(ab - cd) / (f + g)$ إقفز إلى الجملة 75 إذا كانت الكمية موجبة ، وإلى الجملة 260 إذا كانت الكمية مساوية لصفر ، وإلى الجملة 135 إذا كانت الكمية سالبة .

(ب) اطبع التالى على سطر واحد من النهاية الطرفية المركزية "X=" تتبعها قيمة المتغير X ، ثم "Y=" وتتبعها قيمة المتغير Y ، ثم "Z=" وتتبعها قيمة المتغير Z . ابدأ الطباعة عند الأعمدة 4 و 28 و 52 على الترتيب .

(ج) قرر ما إذا كانت قيمة N زوجية أو فردية ، مع فرض أن N لها قسم صحيحة موجبة . (تلميح : قارن قيمة N/2 بالقيمة N/2 بعد بتر الكسر من الرقم) .

(د) فى الجزء (ج) عالية ، ماذا يحدث إذا كانت قيمة N قيمة صحيحة سالبة ؟

٤٢ - هـ يبين كل مثال مما يلى الإشارة إلى متغير أو أكثر من المتغيرات ذات الأدلة . صف نوع المجموعة المتراسة المشار إليها فى كل حالة .

75 LET N\$(3)="ERROR-CHECK" (أ)
20 DI 1 A(12,25),A\$(12,25),B(12),C\$(25) (ب)
100 PRINT P\$(I),P(I,J) (ج)
50 IF Z(J1,J2)<10 THEN 185 (د)

٤٣ - هـ مبين فيما يلى عدة جمل ببسك وجمل متسلسلة تختوى على متغيرات ذات أدلة . بمض الأمثلة مأخوذة بصورة غير صحيحة . تعرف على كل الأخطاء .

50 LET C(I,J)=(3*X+...-2*Y+3)/(17*Z) (أ)
75 LET F(K,5)=Q(K+1,J)+R(K,J+1) (ب)
20 INPUT M,N (ج)
30 DIM A(M,N),X(N),Y(M) (د)
10 DIM K(100),W(10,20),C1,C2,K(100) (هـ)
150 LET S=S+T(K,-3) (و)
200 LET X(K(I))=Y(K(I+1))+Z(K(I-1)) (ز)

٤٤ - هـ اكتب جملة أو أكثر للقيام بكل من العمليات التالية :

(أ) احسب الجذر التربيعى لمجموع مربعات الأعداد ، للعناصر الفردية الأول من القائمة الرقية والتي عددها 100 ، أى ، احسب $[X(1)^2+X(3)^2+X(5)^2+\dots+X(199)^2]^{1/2}$

(ب) احسب عناصر الجدول الرقى H والذي يحتوى على 8 صفوف و 12 عموداً . تقرر قيمة كل عنصر من الجدول H بالصيغة الرياضية :

$$h_{ij} = \frac{1}{i+j-1}$$

(ج) تحتوى قائمة رقية K على عدد N من العناصر . اطبع قيمة كل دليل والعنصر المناظر للعناصر التي لاتتعدى قيمتها 15 . أعرض المخرجات فى عمودين بقيمة الدليل فى العمود الأول والمتغير ذى الدليل المناظر فى العمود الثالث .. اعط عنواناً لكل عمود . ابدأ العمود الأول من المخرجات فى الموضع رقم 8 . (يشير رقم الموضع هنا إلى مكان الطباعة على الورق) ، والعمود الثانى من المخرجات فى الموضع رقم 44 .

(د) جدول رقى W به عدد K من الصفوف و K من الأعمدة احسب حاصل ضرب حدود القطر الرئيسى للجدول W ، حيث يتغير القطر الرئيسى من اليسار الأعلى إلى اليمين الأسفل ، أى ، احسب :

$$W(1,1)*W(2,2)*W(3,3)*\dots*W(K,K).$$

(أ) اطبع عناصر العمود الرابع من الجدول الحرفي M\$ إعرض المخرجات في صورة عمود ، مبتدئاً في الموضع رقم 10 (يشير رقم الموضع هنا إلى مكان الطباعة على الورق) . إفرض أن M\$ تحتوى على عدد M من الصفوف .

(و) كرر المسألة (أ) عاليه مع عرض المخرجات في شكل صف مع ترك مسافة واحدة بين كل عنصر .

(ز) إطبع العناصر في الصف الخامس من الجدول الحرفي M\$ إعرض المخرجات في شكل صف مع ترك مسافة واحدة بين كل عنصر . إفرض أن M\$ تحتوى على عدد N من الأعمدة .

٤٥ - اكتب جمل READ و DATA تلائم كل موقف موصوف أدناه :

(أ) تحدد القيم التالية للقائمة L\$ والمتغيرات P و Q و R و H\$ والجدول الرقبي T

L\$(1)=WHITE	P=2.25E+5	T(1,1)=1	T(2,1)=-2
L\$(2)=YELLOW	Q=6.08E-9	T(1,2)=-3	T(2,2)=4
L\$(3)=ORANGE	R=-1.29E+12	T(1,3)=5	T(2,3)=-6
L\$(4)=RED	H\$=RESTART	T(1,4)=-7	T(2,4)=8

اطبع قائمة بالمتغيرات ذات الأدلة منفصلة في جمل READ .

(ب) كرر الجزء (أ) عاليه مستخدماً الحلقات FOR-TO لعناصر المجموعة المتراسة .

(ج) كرر الجزء (أ) عاليه ، مستخدماً الحلقات FOR-TO لعناصر المجموعة المتراسة ، كما في الجزء (ب) . وعند جزء لاحق من البرنامج أعد المؤشر الرقبي ثم حدد القيم $2.25 E + 5$ و $6.08 E - 9$ و $-1.29 E + 12$ إلى المتغيرات P1 و Q1 و R1 .

(د) كرر الجزء (أ) عاليه ، مستخدماً حلقات FOR-TO لعناصر المجموعة المتراسة كما في شكل (ب) . وعند جزء لاحق من البرنامج ، أعد المؤشر ليحدد القيم الحرفية WHITE و YELLOW و ORANGE و RED للمتغيرات A1\$ و A2\$ و A3\$ و A4\$.

مسائل للبرمجة

Programming Problems

٤٦ - غير البرنامج المبين في مثال ٥ - ٥ بحيث لا يحسب اللوغاريتم عندما تكون قيمة x مساوية للصفر . ضمن البرنامج احتياطي لطباعة 8 نجوم متتالية للوغاريتم صفر . وبذلك يشير إلى حالة طلع .

٤٧ - اكتب برنامج يبسك مائل للمثال ٥ - ٥ لتولد جدول من x و $\sin^2 x$ و $\cos^2 x$ و $\tan^2 x$ و $\cot^2 x$ و $\sec^2 x$ و $\operatorname{cosec}^2 x$ (لاحظ أن $\sec x = 1/\cos x$ و $\operatorname{cosec} x = 1/\sin x$) أنشأ عدد 101 من المداخل لقيم متباعدة بالتساوى حيث تراوح قيم x ما بين صفر و π (وهذا يعنى : اجعل قيمة x مساوية 0 و $\pi/100$ و $2\pi/100$ و \dots و $99\pi/100$ و π) تأكد من أن كل المخرجات لها عناوين ملائمة .

٤٨ - اكتب برنامج يبسك ينشئ جدولاً من القيم للمعادلة :

$$y = 2e^{-0.1t} \sin 0.5t$$

حيث تتغير t ما بين 0 و 60 اسمح لحجم الزيادة في t أن تدخل كعامل إدخال .

٥ - ٤٩ : وسع برنامج الكلمات غير المترجمة في مثال ٥ - ٩ بحيث يطبق كل التوافقيات الممكنة من حرفين أو ثلاثة حروف أو أربعة حروف وذلك لأى أربعة حروف معطاة . اختر مجموعة من أربعة حروف ثم نفذ البرنامج . تعرف على كل الكلمات الإنجليزية الصالحة وذلك بفحص المخرجات مرثياً . هل يمكنك عمل ذلك بدون تشغيل البرنامج . كم عدد التوافقيات المختلفة التى يمكن توليدها من حرفين أو أكثر ؟

٥ - ٥٠ : وسع البرنامج في المثال ٥ - ١٤ والذي يمكن به إعادة تنظيم قائمة من الأرقام بأى طريقة من الطرق الأربع التالية :

(أ) من الأصغر إلى الأكبر جبرياً (من أكبر قيمة سالبة إلى أكبر قيمة موجبة) .

(ب) من الأصغر إلى الأكبر في القيمة . (وذلك بإغفال الإشارات) .

(ج) من الأكبر إلى الأصغر جبرياً .

(د) من الأكبر إلى الأصغر في القيمة .

(لاحظ أنه ليس من المهم أن تكون عناصر القائمة بها قيا موجبة) .

اكتب البرنامج بطريقة تمكنه من القيام بعمل إعادة تنظيم واحدة فقط في كل مرة ينفذ فيها البرنامج . ضمن في البرنامج متغيراً يمكن إدخال قيمته من خلال جملة INPUT في كل مرة ينفذ فيها البرنامج . اجعل إعادة تنظيم القائمة من الأرقام يمكن تقريرها بواسطة القيمة التى تعطى لهذا المتغير (مثال ، إذا كانت $A = 1$ فإن إعادة التنظيم ستكون من الأصغر للأكبر جبرياً ، وإذا كانت $A = 2$ فإن إعادة التنظيم ستكون من الأصغر للأكبر في القيمة ، . . . الخ)

استخدم البرنامج لإعادة تنظيم الأرقام المعطاة في الجدول ٥ - ٣ . أعد تنظيم الأرقام بالطرق الأربع .

جدول ٥ - ٣

43	-85	-4	65
-83	10	-71	-59
61	-51	-45	-32
14	49	19	23
-94	-34	-50	86

٥ - ٥١ : اكتب برنامج يبسك يعيد تنظيم قائمة من الكلمات بترتيب هجائى . ولعمل ذلك ، أدخل الكلمات في قائمة ، حيث يمثل كل عنصر كلمة واحدة . بعد ذلك يمكن ترتيب قائمة الحروف أبجدياً بنفس الطريقة التى يمكن بها إعادة تنظيم قائمة من الأرقام من الأصغر إلى الأكبر (أنظر مثال ٥ - ١٤) .

استخدم البرنامج ليعيد تنظيم الأسماء المعطاة في جدول ٥ - ٤ . كن حريصاً في التعامل مع الحروف الأولى .

جدول ٥ - ٤

Washington	Arthur
Adams, J.	Cleveland
Jefferson	Harrison, B.
Madison	McKinley
Monroe	Roosevelt, T.
Adams, J. Q.	Taft
Jackson	Wilson
Van Buren	Harding
Harrison, W. H.	Coolidge
Tyler	Hoover
Polk	Roosevelt, F. D.
Taylor	Truman
Fillmore	Eisenhower .
Pierce	Kennedy
Buchanan	Johnson, L. B.
Lincoln	Nixon
Johnson, A.	Ford
Grant	Carter
Hayes	Reagan
Garfield	

- ٥٢ - ٥ أعد كتابة برنامج ببسك في مثال ٥ - ١٥ بحيث يحسب حاصل ضرب العناصر في كل صف وفي كل عمود .
- ٥٣ - ٥ اكتب برنامج ببسك يولد جدولاً من معاملات الأرباح المركبة ، F/P حيث :
- $$F/P = [1 + (i/100)]^n$$
- وتمثل i في هذه الصيغة الرياضية نسبة الربح السنوي معبراً عنها كنسبة مئوية ، وتمثل n عدد السنوات .
- اجعل كل صف في الجدول يناظر قيمة مختلفة من n ، وذلك لتراوح n من 1 إلى 30 (من ثم 30 صفاً) .
- واجعل كل عمود يمثل نسبة ربح مختلفة ضمن نسب الأرباح التالية :
- 4 و 4.5 و 5 و 5.5 و 6 و 6.5 و 7 و 7.5 و 8 و 8.5 و 9 و 9.5 و 10 و 11 و 12 و 15 في المائة (من ثم 16 عموداً) . تأكد من إعطاء عناوين مناسبة لكل من الأعمدة والصفوف .
- ٥٤ - ٥ اكتب برنامج ببسك (BASIC) يقرأ مجموعة من درجات الحرارة ، ثم يحسب متوسطها ثم بعد ذلك يحسب الانحراف كل درجة حرارة عن المتوسط .
- يعرف الانحراف كالتالي :
- $$D = T(I) - A$$
- حيث تمثل A متوسط درجة الحرارة . لاحظ أن الانحراف سيكون موجباً إذا كانت درجة الحرارة أعلى من المتوسط ، وسوف يكون سالباً إذا كانت درجة الحرارة أقل من المتوسط .
- اطبع متوسط درجة الحرارة ، تتبعها ثلاثة أعمدة تحتوي على القيم I و $T(I)$ و D على الترتيب . وتأكد من أن كل شيء له عنوان واضح .
- اختبر البرنامج باستخدام مجموعة درجات الحرارة التالية : 28.2 و 29.3 و 33.7 و 42.0 و 58.4 و 71.3 و 84.1 و 83.8 و 74.5 و 53.9 و 41.6 و 34.4
- ٥٥ - ٥ وسع برنامج حساب متوسطات درجات طالب (المسألة ٤ - ٨) بحيث يمكن حساب انحراف متوسط كل طالب عن المتوسط العام للفصل . اطبع متوسط الفصل ، يتبعه إسم الطالب ، ثم درجات الامتحان ، والدرجة النهائية والانحراف عن متوسط الفصل . تأكد أنه تم تنظيم المخرجات منطقياً وتم إعطاؤها عناوين واضحة .
- ٥٦ - ٥ إدرس القائمة التالية لبعض الدول وعاصمتها .

Canada	Ottawa
England	London
France	Paris
India	New Delhi
Israel	Jerusalem
Italy	Rome
Japan	Tokyo
Mexico	Mexico City
People's Republic of China	Peking
United States	Washington
U.S.S.R.	Moscow
West Germany	Bonn

- ٥٧ - ٥ اكتب برنامج ببسك من النوع التخططي والذي سوف يقبل إسم الدولة كمدخل ويعطيك العاصمة المناظرة لها والمكس صحيح . عدة مسائل فنية من أنواع مختلفة تم توصيفها أدناه . جهز مخططاً تمهيدياً مفصلاً وخريطة سير عمليات مناظرة ثم برنامج ببسك كامل لكل من هذه المسائل .

(أ) نفرض أن لدينا جدولاً رقياً A له M صف و N عمود ولدينا أيضاً قائمة رقمية لها N عنصر . والمطلوب توليد قائمة رقمية Y وذلك بالقيام بالعمليات التالية .

$$\begin{aligned} Y(1) &= A(1,1)*X(1)+A(1,2)*X(2)+\dots+A(1,N)*X(N) \\ Y(2) &= A(2,1)*X(1)+A(2,2)*X(2)+\dots+A(2,N)*X(N) \\ &\vdots \\ Y(M) &= A(M,1)*X(1)+A(M,2)*X(2)+\dots+A(M,N)*X(N) \end{aligned}$$

اطبع المدخلات (أى قيم عناصر الجدول A والقائمة X) ، تتبعها القيم المحسوبة لعناصر القائمة Y .

استخدم البرنامج لمعالجة مجموعة البيانات التالية :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{bmatrix}$$

$$X = \begin{bmatrix} 1 \\ -8 \\ 3 \\ -6 \\ 5 \\ -4 \\ 7 \\ -2 \end{bmatrix}$$

(ب) نفرض أن A جدول رقى له K صف و M عمود ، وأن B جدول رقى آخر له M صف و N عمود والمطلوب حساب عناصر جدول رقى C حيث يمكن تحديد كل عنصر فى الجدول بواسطة :

$$C(I,J)=A(I,1)*B(1,J)+A(I,2)*B(2,J)+\dots+A(I,M)*B(M,J)$$

وذلك لتقسيم I تراوح من 1 و 2 و... إلى K ولتقسيم J تراوح من 1 و 2 و... إلى N اطبع عناصر الجداول الرقية A و B و C .

استخدم البرنامج لمعالجة مجموعة البيانات التالية :

$$A = \begin{bmatrix} 2 & -1/3 & 0 & 2/3 & 4 \\ 1/2 & 3/2 & 4 & -2 & 1 \\ 0 & 3 & -9/7 & 6/7 & 4/3 \end{bmatrix}$$

$$B = \begin{bmatrix} 6/5 & 0 & -2 & 1/3 \\ 5 & 7/2 & 3/4 & -3/2 \\ 0 & -1 & 1 & 0 \\ 9/2 & 3/7 & -3 & 3 \\ 4 & -1/2 & 0 & 3/4 \end{bmatrix}$$

(ج) يمكن حساب متعددة الحدود (لجندر) *legendre* بواسطة الصيغ الرياضية التالية :

$$\begin{aligned} P_0 &= 1 \\ P_1 &= x \\ \dots \\ P_n &= \left(\frac{2n-1}{n}\right)xP_{n-1} - \left(\frac{n-1}{n}\right)P_{n-2} \end{aligned}$$

حيث تأخذ n القيم 2 و 3 و 4 و وأن x أى رقم بين -1 و +1

اكتب برنامج بيسك يولد جدول للقيم P_n مقابل قيم x لأى قيمة n من 1 إلى 10 . ولد 201 قيمة من P_n فى كل جدول مبنية على أساس قيم متساوية التباعد للمتغير x (أى اجعل x تأخذ القيم -1.00 و -0.99 و -0.98 و و 0 و 0.01 و 0.02 و و 0.98 و 0.99 و 1.00) واستخرج النتائج فى صورة عمود واضح ومقروء .

(د) إدرس تسلسلا لأرقام حقيقية x_i حيث تأخذ i القيم 1 و 2 و و M . يعرف المتوسط كالتالى :

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_M}{M}$$

الانحراف عن المتوسط هو

$$d_i = (x_i - \bar{x})$$

ومعيار الانحراف هو

$$\sigma = \left[\frac{d_1^2 + d_2^2 + \dots + d_M^2}{M} \right]^{1/2}$$

اقرأ أول M عنصر من مجموعة ملاحظة لها بعد واحد . ثم احسب مجموع هذه العناصر ، والانحرافات ومعيار الانحراف ، والقيمة القصوى الجبرية والقيمة الصغرى الجبرية . طبق هذا البرنامج على بيانات درجات الحرارة المعطاة فى المسألة ه - ٤ هـ

كرر العمل لعدد K من المجموعات المتراسة المختلفة . ثم احسب المتوسط العام ومعيار الانحراف العام والقيمة المطلقة للحد الأقصى (الأكبر) والقيمة المطلقة للحد الأدنى (الأصغر جبريا) .

(هـ) اكتب برنامج بيسك لحساب التباوت \bar{v} ، لقائمة من الأرقام بطريقتين باستخدام الصيغ الرياضية :

$$\bar{v} = \frac{1}{M} [(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_M - \bar{x})^2]$$

$$\bar{v} = \frac{1}{M} (x_1^2 + x_2^2 + \dots + x_M^2) - \bar{x}^2$$

وفى هذه الصيغ الرياضية تحسب قيمة المتوسط \bar{x} كما فى الصيغة الرياضية :

$$\bar{x} = \frac{1}{M} (x_1 + x_2 + \dots + x_M)$$

حيث M هى عدد القيم فى القائمة .

رياضياً ، يمكن رؤية أن الصيغتين الرياضيتين لحساب \bar{v} متماثلتان عندما تكون الأرقام المعطاة لها قيم متقاربة جداً من بعضها ، ومن ثم فإن القيمة التى نحصل عليها \bar{v} من استخدام الصيغة الرياضية الثانية اعتبارياً خطأ . والسبب فى ذلك أننا يجب أن نحسب الفرق بين قيمتين تقريباً متساويتين . مثل هذه الفروق المحسوبة يمكن أن تكون عدم دقتها عالية . وتنتج الصيغة الرياضية الأولى نتائج أكثر دقة تحت هذه الظروف .

وضح أن الجملة السابقة صحيحة وذلك بحساب اختلاف البيانات المعطاة فى جدول ه - هـ (القيمة الصحيحة هى : $\bar{v} = 0.00339966$)

جدول ٥ - ٥

99.944	100.054	100.059	100.061
100.039	100.066	100.029	100.098
99.960	99.936	100.085	100.038
100.033	99.932	100.079	100.024
99.993	99.913	100.095	100.046

لاحظ : أن هذا المثال يبرهن على عدم صحة الاعتقاد الشائع والخاص أن الحاسب دائماً ما ينتج إجابات صحيحة جداً .

(و) تحسب تكاليف رهن منزل بطريقة يفترض فيها أن يدفع مبلغ ثابت شهرياً خلال مدة الرهن للهيئة المقرضة . إلا أن الجزء من إجمالي المدفوعات الشهرية الذى يمثل الفوائد على الرصيد غير المدفوع من القرض يختلف من شهر لآخر . خلال مدة الرهن الأولى . تمثل معظم المدفوعات الشهرية المبالغ المطلوبة لتسديد الفوائد على الديون وكسر صغير فقط من المدفوعات الشهرية يمثل تسديدات تدريجية للقرض ، وبذا يتناقص الرصيد غير المدفوع تدريجياً ، أى يرتب على ذلك أن تتناقص مدفوعات الفوائد الشهرية بينما يزداد المبلغ الموجه لتسديد الدين الأصل ، وبذلك فإن رصيد القرض ينخفض بنسبة متصاعدة .

يعلم تماماً المزمع على شراء منزل المبلغ الذى يطلب اقتراضه والوقت اللازم لتسديد الديون . ثم يسأل الهيئة المقرضة بعد ذلك عن الأقساط الشهرية حسب سعر الفائدة السائد . ويجب أن يهتم أيضاً بمقدار الفوائد التى تخصم من المدفوعات الشهرية ، وإجمالي الفوائد التى تم دفعها من أول القرض ، والمبالغ المتبقية عليه من أصل الدين والواجب دفعها للهيئة المقرضة عند نهاية كل شهر .

اكتب برنامج بيسك يمكن أن يستخدم بواسطة الهيئة المقرضة ليمد العميل بكل هذه المعلومات . نفرض أنه سبق توصيف قيمة القرض ، نسبة الربح السنوية ومدة القرض وأن المدفوعات الشهرية تحسب بواسطة الصيغة الرياضية :

$$A = iP \left[\frac{(1+i)^n}{(1+i)^n - 1} \right]$$

حيث A هي المدفوعات الشهرية بالدولار .

P إجمالي القرض بالدولار .

i نسبة الربح الشهرى معبراً عنه برقم كسرى ، (مثال $1\frac{1}{2}\%$ يجب كتابتها 0.005) .

n عدد الأقساط الشهرية .

ويمكن حساب الفائدة الشهرى المدفوعة من الصيغة الرياضية .

$$I = iB$$

حيث I فوائد المدفوعات الشهرية بالدولار .

B الرصيد غير المدفوع بالدولار .

ويساوى الرصيد غير المدفوع ببساطة القرض الأصل مطروحاً منه المدفوعات السابقة لتسديد الدين الأصل . المدفوعات الشهرية لتسديد الدين الأصل ، أى الكمية المستخدمة لتخفيض الرصيد غير المدفوع ، هى ببساطة

$$T = A - I$$

حيث T هي المدفوعات الشهرية لتسديد الدين الأصل .

استخدم البرنامج لحساب تكاليف قرض مقداره \$30,000 ولمدة 25 سنة ونسبة فوائد سنوية 8% ثم بعد ذلك كرر الحسابات لنسبة فائدة سنوية 8.5% . ما هو الأثر الفعلي على قيمة القرض نتيجة زيادة الفائدة بمقدار 0.5% طوال مدة الرهن ؟

(ز) تعرف الطريقة المستخدمة لحساب تكلفة رهن منزل في المسألة ٥ - ٥٧ سابقاً بطريقة المدفوعات الثابتة ، حيث أن أقساط المدفوعات الشهرية ثابتة . وإذا فرضنا بدلاً من ذلك أن المدفوعات الشهرية تم حسابها بطريقة الربح البسيط . أي بفرض أن تسديد القرض يتم بأقساط شهرية ثابتة قيمتها :

$$T = P/n$$

وبالإضافة إلى ذلك ، تسدد الفائدة على الرصيد غير المسدد بمقدار ثابت شهرياً فان :

$$I = iB$$

وبذلك يكون المبلغ المسدد شهرياً (A) هو حاصل جمع T و I أي $A = T + I$ ويتناقص كل شهر كلما تلاشى الرصيد غير المسدد .

اكتب برنامج يسك لحساب تكلفة رهن منزل باستخدام هذه الطريقة في السداد . عنون المخرجات بوضوح . واستخدم البرنامج لحساب تكلفة رهن مقداره \$30,000 لمدة 25 سنة ونسبة فائدة 8% سنوياً . قارن هذه النتائج بالنتائج التي تم الحصول عليها من المسألة ٥ - ٥٧ (و) .

(ح) نفرض أننا أعطينا مجموعة من القيم المحدولة لكل من y مقابل x أي :

y_0	y_1	y_2	\dots	y_n
x_0	x_1	x_2	\dots	x_n

ونرغب في الحصول على قيمة y عند أي قيمة x تقع ما بين قيمتين في الجدول . والطريقة الشائعة لحل هذه المسألة هي طريقة الاستكمال أي بتمرير كثيرة الحدود $y(x)$ خلال n نقطة حيث :

$$y(x_0) = y_0, y(x_1) = y_1, \dots, y(x_n) = y_n$$

والطريقة الشائعة للقيام بهذا الاستكمال هي استخدام صيغة لاجرانج *lagrange* لاستكمال كثيرة الحدود . ولعمل ذلك نكتب :

$$y(x) = f_0(x) \cdot y_0 + f_1(x) \cdot y_1 + \dots + f_n(x) \cdot y_n$$

حيث أن $f_i(x)$ كثيرة الحدود بحيث

$$f_i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

لاحظ أن $f_i(x_i) = 1$ و $f_i(x_j) = 0$ حيث x_j قيمة x في الجدول وتختلف عن x_i . ولذلك فأننا نؤكد أن

$$y(x_i) = y_i$$

اكتب برنامج يسك ليقرأ أزواجاً من البيانات مقدارها n حيث لا تتعدى n القيمة 10 ، ثم بعد ذلك احصل على قيمة مستقلة للمتغير y عند نقطة محددة أو أكثر من قيم x . استخدم البرنامج لتحصل على قيم مستقلة للمتغير y عند $x = 13.7$ و $x = 37.2$ و $x = 112$ و $x = 147$ من البيانات الموجودة في الجدول ٥ - ٦ . وقرر كم عدد أزواج البيانات المحدولة والمطلوب في آرن الحسابات من أجل الحصول على قيمة مستقلة وصحيحة ومعقولة للمتغير y .

جدول ٥ - ٦

y	0.21073	0.37764	0.45482	0.49011	0.50563	0.49245	0.47220	0.43433	0.33824	0.19390
x	0	10	20	30	40	50	60	80	120	180

(ط) يصف المثال ٤ - ٥ طريقة التعويض المتعاقب لحل معادلة جبرية للصيغة $x = F(x)$ بواسطة طريقة فنية للتكرار تستخدم الصيغة التكرارية $x_{i+1} = F(x_i)$.

طريقة أخرى ، وغالباً ما تكون أكثر فعالية ، لحل المعادلات من هذا الطراز وهي تكرار نيوتن - رابسون *Newton - Raphson* (وتسمى أحياناً طريقة نيوتن) ولاستخدام هذه الطريقة ، يجب أن تكتب المعادلات الجبرية في الصيغة $f(x) = 0$ وتستخدم الصيغة التكرارية :

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

حيث تمثل $f'(x_i)$ المشتقة الأولى من $f(x)$ محسوبة عند x_i .

تنفذ التكرارات بنفس طريقة التعويضات المتعاقبة أي تحسب قيمة x_{i+1} من الصيغة العكسية ثم تقارن بقيمة x_i . إن لم تكن القيمتان متقاربتين تقارباً كافياً ، تستخدم قيمة x_{i+1} للتعويض في الجانب الأيمن من المعادلة التكرارية وتكرر الحسابات مرة أخرى .

اكتب برنامج ببسك لحل معادلات جبرية غير خطية بأى من الطرق الفنية السابقة . قرر أى طريقة سوف تستخدم وذلك بتحديد قيمة رقمية مناسبة لمتغير إدخال .

استخدم البرنامج لحل المعادلة $x + \cos x = 1 + \sin x$ لبعض قيم x التي تتراوح قيمها ما بين $\pi/2$ و π . حل باستخدام كلتا الطريقتين الفئيتين . أى طريقة تبدو أفضل ؟

(ى) نفرض أننا أعطينا عدداً من النقط المنفصلة (x_1, y_1) و (x_2, y_2) و و (x_n, y_n) حيث نقرأ من المنحنى $y = f(x)$ ، وتراوح قيمة x ما بين x_1 و x_n ونرغب في حساب المساحة التقديرية الواقعة تحت المنحنى وذلك بتقسيم المنحنى إلى عدد من المستطيلات الصغيرة وحساب مساحات هذه المستطيلات . (وتعرف هذه بقاعدة شبه المنحرف *Trapezoidal rule*) استخدم الصيغة الرياضية :

$$A = \frac{1}{2}(y_1 + y_2)(x_2 - x_1) + \frac{1}{2}(y_2 + y_3)(x_3 - x_2) + \dots + \frac{1}{2}(y_{n-1} + y_n)(x_n - x_{n-1})$$

لاحظ أن متوسط الارتفاع لكل مستطيل يعطى بواسطة $\frac{1}{2}(y_i + y_{i+1})$ وأن العرض لكل مستطيل مساوياً لقيمة $(x_{i+1} - x_i)$ حيث تأخذ i القيم 1 و 2 و و $(n-1)$.

استخدم البرنامج لحساب المساحة تحت المنحنى $y = x^3$ بين حدود $x = 1$ و $x = 4$. حل هذه المسألة أولاً بعدد 16 من النقط المتساوية التباعد ، ثم بعد ذلك بعدد 61 نقطة وأخيراً بعدد 301 نقطة - لاحظ أنسوة الحل سوف تأخذ في التحسن كلما زاد عدد النقط . (الإجابة الصحيحة لهذه المسألة هي 63.75) .

(ك) تصف المسألة ٥ - ٥٧ (ى) عاليه ، طريقة تعرف بقاعدة شبه المنحرف *Trapezoidal rule* لحساب المساحة تحت المنحنى $y(x)$ ، وحيث تستخدم مجموعة من القيم المجدولة (x_1, y_1) و (x_2, y_2) و و (x_n, y_n) وذلك لوصف المنحنى . إذا كانت القيم المجدولة متساوية التباعد فإن المعادلة المعطاة في ٥ - ٥٧ (ى) يمكن تبسيطها لقراءة :

$$A = \frac{1}{2}(y_1 + 2y_2 + 2y_3 + 2y_4 + \dots + 2y_{n-1} + y_n) \Delta x$$

حيث Δx هي المسافة ما بين قيمتين متعاقبتين للمتغير x .

طريقة فنية أخرى يمكن تطبيقها عندما يوجد عدد زوجي من المسافات متساوية التباعد ، أى عدداً فردياً لنقط البيانات ، وتعرف بقاعدة سيمبسون (Simpson rule) . ومعادلة الحسابات لتطبيق قاعدة سيمبسون هي :

$$A = \frac{1}{3}(y_1 + 4y_2 + 2y_3 + 4y_4 + 2y_5 + \dots + 4y_{n-1} + y_n) \Delta x$$

وبالنسبة لأي قيمة معينة من Δx . سوف تنتج قاعدة سيمبسون نتائج أكثر دقة عن تلك التي تنتج من قاعدة شبه المنحرف .

اكتب برنامج ببسك لحساب المساحة تحت منحنى باستخدام كل من الطرق السابقة ، مع فرض رقم فردى من نقط البيانات متساوية التباعد . ثم قرر أيًا من هذه الطرق سوف تستخدم وذلك باعطاء قيمة رقمية مناسبة لبعض متغيرات الإدخال . اسمح بعدد يصل إلى 101 مجموعة من البيانات ، حيث يمكن لقيم البيانات المحدولة أن تقرأ للحاسب أو أن تحسب داخلياً باستخدام معادلة جبرية .

استخدم البرنامج لحساب المساحة تحت المنحنى :

$$y = e^{-x^2}$$

حيث تراوح قيم x من صفر إلى واحد . احسب المساحة باستخدام كل من طريقتي الحسابات ، وقارن بين النتائج بالإجابة الصحيحة وهي $A = 0.7468241$.

(ل) يمكن كتابة المعادلة التفاضلية من الدرجة الأولى كما يلي :

$$\frac{dy}{dx} = f(x, y), \quad x \geq x_0$$

مع $y(x_0) = y_0$ حيث y_0 تمثل قيمة رقمية معروفة . والهدف من حل المعادلة التفاضلية هو الحصول على معادلة ، أو مجموعة من القيم المحدولة للمتغير y كدالة في المتغير x .

يمكن حل معادلة تفاضلية من هذا النوع بواسطة « خطوات متقدمة » متتالية بمسافات صغيرة في اتجاه x . وبمعنى آخر ، ابدأ بالقيمة المعروفة $y(x_0)$ ، يمكن حساب قيمة $y(x_1)$ حيث $x_1 = x_0 + \Delta x$ و Δx رقم صغير معرف سابقاً (حجم الخطوة) . ثم بعد ذلك يمكن الحصول على قيمة $y(x_2)$ ، حيث $x_2 = x_1 + \Delta x$ ؛ ثم $y(x_3)$ حيث $x_3 = x_2 + \Delta x$ وهكذا حتى يتم حساب عدد كاف من النقط .

أسهل طريق لحساب قيمة $y(x_i + 1)$ عند إعطاء قيمة $y(x_i)$ هو استخدام طريقة أويلر (Euler's method)

$$y_{i+1} = y_i + f(x_i, y_i) \Delta x$$

حيث y_i تمثل $y(x_i)$ و y_{i+1} تمثل $y(x_i + 1)$ و $f(x_i, y_i)$ تمثل dy/dx محسوبة عند (x_i, y_i) . طريقة أويلر طريقة سهلة للعمل بها ، ولكن نتائجها إلى حد ما هي تقريب غير صحيح للمنحنى $y(x)$ إلا إذا اختيرت Δx على أنها قيم صغيرة جداً . على ذلك يتطلب عدداً كبيراً من النقط من أجل حساب $y(x)$ على مدى واسع نسبياً للمتغير ، $x_0 \leq x \leq x_n$ ،

هناك طريقة أكثر دقة لحساب قيمة y_{i+1} تعرف بطريقة رونج كوتا (Runge Kutta method) من الدرجة الرابعة :

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(x_i, y_i) \Delta x$$

$$k_2 = f(x_i + \Delta x/2, y_i + k_1/2) \Delta x$$

$$k_3 = f(x_i + \Delta x/2, y_i + k_2/2) \Delta x$$

$$k_4 = f(x_i + \Delta x, y_i + k_3) \Delta x$$

حيث

وعلى ذلك بالنسبة لنقطة محددة (x_i, y_i) تتلخص الطريقة في حساب $x_{i+1} = x_i + \Delta x$ ومنها يتم حساب قيم للمتغيرات k_1 و k_2 و k_3 و k_4 وأخيراً تحدد قيمة y_{i+1} .

اكتب برنامج ييسر حل معادلة تفاضلية من الدرجة الأولى بشرط ابتدائي محدد باستخدام إحدى الطريقتين السابقتين. ثم قرر أيًا من الطريقتين سوف تستخدم باعطاء قيمة رقمية مناسبة لأحد المتغيرات. صف حجم الخطوة (Δx) والعدد الإجمالي للخطوات (n) كدخلات.

استخدم البرنامج في حل المعادلة التفاضلية :

$$\frac{dy}{dx} = x - y, \quad 0 \leq x \leq 2$$

حيث $y(0) = 1$ ، أوجد حلاً باستخدام إحدى الطريقتين السابقتين. قارن النتائج التي تم الحصول عليها بالنتائج الصحيح والمطاة بالمعادلة $y = 2e^{-x} + x - 1$. قرر إلى أي حد يجب أن يكون حجم الخطوة صغيراً لكل طريقة من أجل الحصول على حل صحيح إلى ثلاثة أرقام معنوية.

الجزء الثاني : البيسك المتقدم

الفصل ٦

الدوال والبرامج الفرعية الصغيرة Functions and Subroutines

٦ - ١ تعريف الدالة — جملة DEF

DEFINING A FUNCTION—THE DEF STATEMENT

لتجنب تكرار برمجة نفس الحسابات ، يمكن أن يكتب المبرمج الدوال الخاصة به ، لاستخدامها بجانب الدوال المكتيبة . تعرف الدالة ذات السطر الواحد بواسطة جملة **DEF** (**DEFINE**) . وتتكون الجملة من رقم السطر ، ثم الكلمة الدالة **DEF** ثم تعريف الدالة . ويتكون تعريف الدالة نفسها من اسم للدالة يتبعه علامة التساوى ثم يتبعه ثابت مناسب أو متغير مناسب أو صيغة رياضية . وإذا تطلبت الدالة خلاصات فيجب أن تظهر فوراً بعد اسم الدالة ، محصورة بين قوسين ومنفصلة عن بعضها بواسطة فاصلات (,) . ويسمح باستخدام المتغيرات التي ليس لها أدلة فقط كخلاصات في تعريف الدالة .

يمكن أن تعرف كل من الدوال الرقية والدوال الحرفية بجملة **DEF** (تعطى الدالة الرقية قيمة عددية ، أما الدالة الحرفية فتعطي قيمة سلسلة حرفية) . إذا كانت الدالة رقية فيجب أن يتكون اسمها من ثلاثة حروف ، حيث أول حرفين يجب أن يكونا **FN** . وبذلك فيمكن وجود عدد من الدوال يصل إلى 26 دالة منفصلة في البرنامج الواحد (**FNA** و **FNB** و **FNC**.....) .

ويجب أن يتكون اسم الدالة الحرفية من ثلاثة حروف ويتبعها علامة الدولار (\$) ومرة أخرى يجب أن يكون أول حرفين هما **FN** . كذا يمكن تعريف عدد من الدوال الحرفية يصل عددها إلى 26 دالة في البرنامج الواحد (**FNA\$** و **FNB\$** و **FNC\$**.....) . الدالة الحرفية والرقية التي لها نفس الحروف الثلاثة (مثال **FNP\$** و **FNP**) كل منها كيان منفصل ويمكن أن تظهر في نفس البرنامج .

مثال ٦ - ١

مبين فيما يلي ثلاثة تعريفات نموذجية لدوال من سطر واحد :

```
10 DEF FNA(X)=X↑3+2*X↑2-3*X+4
20 DEF FNC$="NAME AND ADDRESS:"
30 DEF FNR(A,B,C)=SQR(A↑2+B↑2+C↑2)
```

تعرف الجملة الأولى والثالثة دوال رقية **FNA** و **FNR** ، وتعرف الجملة الثانية دالة حروف **FNC\$** . لاحظ أن الدالة الثانية لا تحتوي على خلاصة . ولاحظ أيضاً أن الدالة الثالثة تستخدم الدالة المكتيبة **SQR** في تعريف الدالة .

تتطلب بعض نسخ البيسك أن تسبق تعريف الدالة ما يناظرها من مراجع دالة ، وتسمح نسخ أخرى بظهور تعريف الدالة في أى مكان في برنامج البيسك وفي كلتا الحالتين . فالأسلوب الجيد لممارسة البرمجة هو تجميع كل تعاريف الدوال ووضعها بقرب بداية البرنامج ، ويسهم ذلك في تكوين برنامج مرتب واضح ومقروء .

ويجب أن نوضح أن وجود جملة **DEF** تسهم فقط في تعريف الدالة . وحساب قيمة الدالة من المهم أن نشير إلى اسم الدالة في مكان آخر من البرنامج كما نعمل تماماً مع الدالة المكتيبة . وسوف ترى كيف ننجز ذلك في القسم التالى .

٦ - ٢ الإشارة إلى الدالة REFERENCING A FUNCTION

يمكن الإشارة إلى الدالة (أى حساب قيمتها) وذلك بذكر اسم الدالة من خلال جملة بيسك ، كما لو كان اسم الدالة متغيراً عادياً . ويجب أن يتبع اسم الدالة مجموعة من الخلاصات محصورة بين أقواس ومنفصلة بواسطة فاصلات (,) .

عند حساب قيمة الدالة فان قيم الخلاصات تعطى بواسطة إشارة الدالة وليس تعريف الدالة . ولهذا السبب ، فان الخلاصات التي تظهر في جملة DEF تسمى خلاصات زائفة . وأسماء الخلاصات في الإشارة للدالة لا تستلزم أن تكون هي نفسها في تعريف الدالة . بينما يجب أن يكون الخلاصات نفس العدد ، ويجب أن تكون الخلاصات من النوع الصحيح (أى عددي أو حرفي) .

مثال ٦ - ٢

بين فيما يلي تكوين هيكل لبرنامج ببسك يحتوى إشارتين لدالة معرفة بواسطة المبرمج .

```
10 DEF FNA(X)=X+3+2*X+2-3*X+4
...
50 LET W=FNA(Y)+Z
...
90 IF FNA(C)>=C1 THEN 140
```

يترتب على الجملة رقم 50 حساب قيمة الدالة FNA باستخدام القيمة الحالية للمتغير Y كعامل مدخل . (إذن فان الدالة سوف تعطى القيمة $Y+3+2*Y+2-3*Y+4$) وفي الجملة رقم 90 تحسب قيمة نفس الدالة ولكن باستخدام القيمة الحالية C (وبذلك تعطى القيمة $C+3+2*C+2-3*C+4$) .

لاحظ أن تعريف الدالة كان يمكن أن يلي الإشارة للدالة بدلا من أن يسبقها . يجب أن تناظر الخلاصات في مرجع الدالة للخلاصات الزائفة على أساس واحد لواحد (أى تناظر كل خلاصة الخلاصة الزائفة) عندما يراد أكثر من خلاصة واحدة . ومرة أخرى يجب أن يكون التناظر بالنسبة لعدد الخلاصات ونوع كل خلاصة ، ولكن ليس بالنسبة لأسماء الخلاصات .

مثال ٦ - ٣

يحتوى برنامج ببسك على الجمل التالية :

```
80 LET A=FNR(C,X,Y)
...
250 DEF FNR(A,B,C)=SQR(A+2+B+2+C+2)
```

سوف تسبب جملة LET (السطر رقم 80) في حساب قيمة الدالة $SQR(C+2+X+2+Y+2)$ وبعد ذلك تعطى النتيجة للمتغير A .

لاحظ أن الإشارة للدالة (السطر رقم 80) وتعريف الدالة (السطر رقم 250) يحتوى كل على 3 خلاصات رقمية ولكن أسماء الخلاصات لا تتناظر .

وبالرجوع إلى القسم ٦ - ١ تذكر أن أسماء الخلاصات التي تظهر في تعريف الدالة (أى الخلاصات الزائفة) يجب أن تكون متغيرات بدون أدلة . ولكن لنا مطلق الحرية في الإشارة للدالة . فهنا يمكن كتابة الخلاصات كشوايت وكمتغيرات ذات أدلة أو صيغاً رياضية أو إشارات لدوال أخرى . في الحقيقة قيمة كل خلاصة هي التي تستخدم في الحسابات .

مثال ٦ - ٤

يحتوى برنامج ببسك على الجمل التالية :

```
30 DEF FNR(A,B,C)=SQR(A+2+B+2+C+2)
...
170 LET X3=FNR(K(I),5*(P+Q),LOG(T))
```

لاحظ أن تعريف الدالة يحتوي فقط على متغيرات بدون أدلة كخلاصات . ولكن ، في الإشارة إلى الدالة ، نرى أنه يمكن التعبير عن الخلاصات كمتغيرات ذات أدلة أو صيغاً رياضية أو إشارات لدوال مكتوبة . وتنفيذ البرنامج يتسبب في حساب قيمة الدالة .

```
SQR(K(I)+2+(5*(P+Q))+2+LOG(T)+2)
```

وتمثل قيمة الناتج بعد ذلك للمتغير X3 .

لا يستلزم أن تتقيد المتغيرات المستخدمة في تعريف الدالة بالخلاصات . فيمكن لأي متغيرات أخرى للبرنامج أن تظهر أيضاً (متضمنة متغيرات ذات أدلة) . وعند حساب قيمة الدالة فإنها تستخدم القيم الأكثر حداثة التي أعطيت لهذه المتغيرات .

مثال ٦ - ٥

فيما يلي تكوين هيكل البرنامج ببسك .

```
30 DEF FNZ(X,Y)=(C1*X+C2*Y)/(C1+C2)
...
60 LET C1=10
70 LET C2=20
80 LET R=FNZ(P,Q)
```

تنفيذ الجملة رقم 80 سوف يتسبب في حساب قيمة الدالة :

```
(10*P+20*Q)/(10+20)
```

لاحظ أن القيم C1 و C2 لا تمثل خلاصات . وتستخدم القيم الأكثر حداثة التي تم إعطاؤها للمتغيرين C1 و C2 (أي C1 = 10 ، C2 = 20) عند حساب قيمة الدالة .

في مثال ٦ - ٦ التالي نرى توضيحاً مفصلاً لاستخدام دالة معرفة بواسطة البرنامج .

مثال ٦ - ٦ البحث عن أقصى قيمة Search for a Maximum

نفرض أننا نرغب في إيجاد قيمة x التي يترتب عليها أن تزداد الدالة :

$$y = x \cos x$$

إلى الحد الأعلى وذلك خلال المدى المحصور بين $x = 0$ ناحية اليسار و $x = \pi$ ناحية اليمين . وسوف تتطلب معرفة قيمة x بكل دقة . وسوف نطلب أيضاً أن يكون مخطط البحث ذا كفاءة عالية نسبياً بمعنى أنه يجب أن تحسب الدالة $y = x \cos x$ أقل عدد ممكن من المرات .

ولحل هذه المسألة يمكن أن تولد أرقام كثيرة لدوال تجريبية متقاربة للمسافات (وهي حساب قيمة الدالة عند $x=0$ وعند $x=0.0001$ وعند $x=0.0002$ و... و $x=3.1415$ و $x=3.1416$) ثم نحدد أكبر القيم وذلك بالفحص المرفق . ولن تكون كفاءة هذه الطريقة عالية ، لأنها تتطلب تدخلاً إنسانياً للحصول على النتيجة النهائية . وبدلاً من ذلك دعنا نستخدم مخطط الحذف وهو إجراء حساب له فاعلية عالية لكل الدوال والتي لها « نقطة قصوى » واحدة فقط خلال مدى البحث .

طريقة اجراء الحسابات Computational Procedure

نفرض أننا وضعنا نقطتي بحث عند منتصف المدى ، متباعدة بمقدار مسافة صغيرة جداً عن بعضها كما هو مبين في شكل ٦ - ١ ، حيث :

$$X1 = \text{النهاية اليسرى لمدى البحث .}$$

$$X2 = \text{نقطة البحث الداخلية ناحية اليسار .}$$

$$X3 = \text{نقطة البحث الداخلية ناحية اليمين .}$$

$$X3 = \text{النهاية اليمنى لمدى البحث .}$$

$$D = \text{المسافة ما بين } X2 \text{ و } X3 .$$

إذا تم تعريف $X1$ و $X2$ و D فإن النقط الداخلية يمكن حسابها :

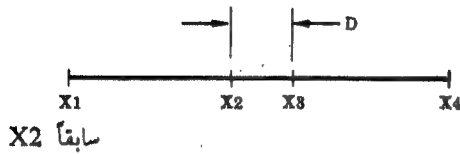
$$X2 = X1 + .5 * (X4 - X1 - D)$$

$$X3 = X1 + .5 * (X4 - X1 + D) = X2 + D$$

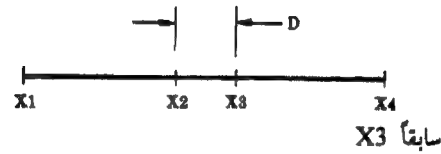


شكل ٦ - ١

دعنا نحسب قيمة الدالة $y = x \cos x$ عند $X2$ وعند $X3$ ودعنا نسمي هذه القيم $Y2$ و $Y3$ على الترتيب . نفرض أن $Y2$ قيمتها أكبر من $Y3$ وبذلك نعلم أن أقصى قيمة نبحث عنها تقع في أي مكان بين $X1$ و $X3$ وبذلك نحفظ هذا الجزء فقط لمدى البحث والذي يتراوح ما بين $x = X1$ إلى $x = X3$ (والآن سوف نشير للنقطة القديمة $X3$ كما لو كانت $X4$ ، حيث أنها النهاية اليمنى لمدى البحث الجديد) ثم نولد نقطتي بحث جديدتين $X2$ و $X3$. وسوف توضع هذه النقط في منتصف مدى البحث القديم ، بينهما المسافة D ، كما هو مبين في شكل ٦ - ٢ .



شكل ٦ - ٣



شكل ٦ - ٢

ومن ناحية أخرى ، نفرض أن قيمة $Y3$ في مدى البحث الأصلي كانت أكبر من قيمة $Y2$ فإن ذلك يشير إلى أي مدى يجب أن يقع البحث الجديد ما بين $X2$ و $X4$ وبذلك فإننا نعيد تسمية النقطة المسماة أصلاً $X2$ بالإسم الجديد $X1$ ونولد نقطتي بحث جديدتين $X2$ و $X3$ عند منتصف مدى البحث الجديد كما هو مبين في شكل ٦ - ٣ .

نستمر في توليد زوج جديد من نقط البحث عند منتصف كل مدى جديد ، ثم نقارن بين قيم Y ، ثم نحذف جزءاً من مدى البحث حتى يصبح المدى أقل من $D * 3$ وعند حدوث ذلك لا يمكننا التفرقة بين النقط الداخلية وبين حدود المدى . من م فأننا قد وصلنا إلى نهاية البحث .

عندما تقارن قيم $Y2$ و $Y3$ في كل مرة نحذف من مدى البحث الجزء الذي يحتوي على القيمة الأصغر للمتغير Y . أما إذا كانت القيمتان الداخليتان للمتغير Y متساويتان (ويمكن أن يحدث ذلك ، بالرغم أنه غير عادي) ، عندئذ سوف يتوقف إجراء البحث ونفترض أن النقطة القصوى تحدث عند منتصف نقطتي البحث .

فور انتهاء البحث ، سواء كان ذلك لأن مدى البحث أصبح صغيراً صغراً كافياً أو لأن النقطتين الداخليتين تعلمان قياً متطابقة لـ Y فيمكننا حساب أقصى موقع بالتقريب على أنه :

$$X5 = 0.5 * (X2 + X3)$$

ويمكن الحصول على القيمة القصوى المناظرة للدالة من $X5 * \cos(X5)$

المخطط التمهيدى للبرنامج The Program Outline

- ١ - عرف الدالة $y = x \cos x$.
- ٢ - اقرأ القيم البدائية للمتغيرين $X1$ و $X2$ والمسافة D .
- ٣ - اجعل قيمة $I = 1$ (حيث I عداد داخل) .
- ٤ - احسب زوجاً من النقط الداخلية .
- ٥ - اكتب قيم x عند نهاية المدى وعند النقط الداخلية واكتب قيم Y المناظرة لها .
- ٦ - قارن بين $X2$ و $X3$.
- (أ) إذا كانت $Y2$ أكبر من $Y3$ ، فبدل أسماء $X3$ و $X4$ ، وبذلك تعرف مدى جديداً للبحث وتقدم للخطوة رقم 7 .
- (ب) إذا كانت $Y3$ أكبر من $Y2$ ، فبدل أسماء $X1$ و $X2$ ، وبذلك تعرف مدى جديداً للبحث وتقدم للخطوة رقم 9 .
- (ج) إذا كانت $Y2$ تساوي $Y3$ تقدم للخطوة رقم 9 .
- ٧ - اختبر لترى ما إذا كانت $I = 100$ وإذا تحقق ذلك اكتب رسالة ملائمة وتوقف . وإلا زد عداد التكرار واستمر .
- ٨ - اختبر لترى ما إذا كانت $(X4 - X1) > 3 * D$ وإذا تحقق ذلك ارجع مرة ثانية للخطوة رقم 4 وإلا أكمل للخطوة التالية .

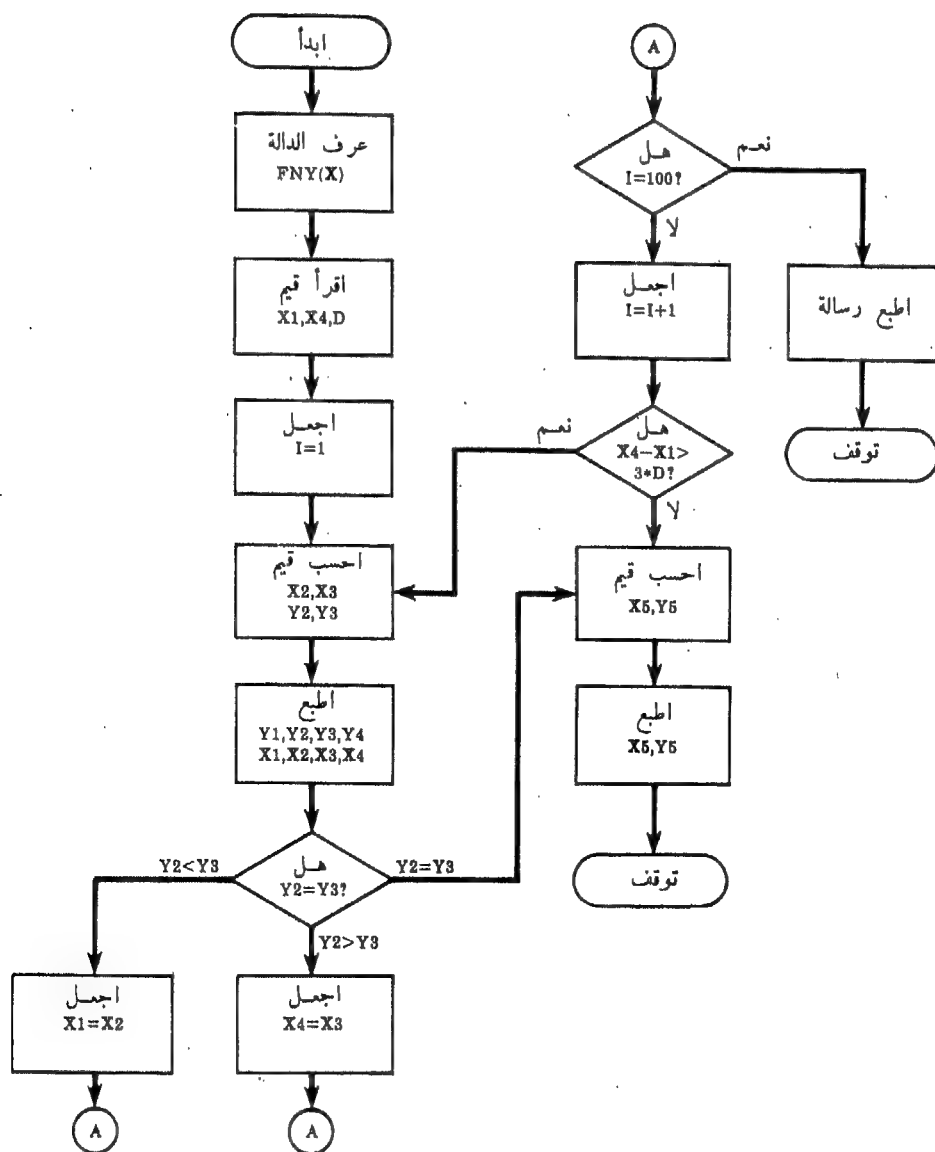
$$X5 = 0.5 * (X2 + X3)$$

٩ - احسب قيمة

$$Y5 = X5 * \cos(X5)$$

وبعد ذلك اكتب النتائج النهائية وتوقف .

مبين في شكل ٦ - ٤ خريطة سير عمليات لهذا الإجراء .



شكل ٦ - ٤

برنامج البيسك The BASIC Program

يظهر برنامج البيسك الحقيقي في شكل ٦ - ٥ . وللتبسيط فأننا ضمننا دالة معرفة لحساب قيمة الكمية $y = x \cos x$ (لاحظ أن تعريف الدالة يتضمن الإشارة إلى الدالة المكتوبة COS). ولاحظ أننا أشرنا إلى الدالة في السطور 170 و 180 و 200 و 440. وتعطى قيما مختلفة للخلاصة في كل إشارة للدالة (أي قيم مختلفة للمتغير x).

ويبين شكل ٦ - ٦ المخرجات المولدة بواسطة البرنامج عندما $X1 = 0$ و $X4 = 3.14159$ و $D = 0.0001$. ورأينا أن أقصى قيمة للمتغير y هي 5.5611 تقريباً وتحدث عند $x = 0.8604$. لاحظ أنه تم الحصول على هذه النتيجة لدرجة عالية من الدقة بعدد من نقاط البحث وصلت إلى 14 قيمة!

ويمكن أيضاً استخدام الإجراء المعطى في هذا المثال للحصول على أقل قيمة للدالة x في الحقيقة. يمكن استخدام البرنامج نفسه بعد إجراء تعديلات طفيفة. تساعدنا هذه الطريقة للحصول على أقل قيمة بطريقة فنية عالية الكفاءة لحساب جذور المعادلة الجبرية غير الخطية. فمثلاً، نفرض أننا نريد إيجاد قيمة x التي تجعل $f(x)$ مساوية للصفر. أحد الأمثلة لذلك: $f(x) = x - \sin x + \cos x - 1$. إذا جعلنا $y(x) = f(x)^2$ فإن الدالة $y(x)$ سوف تكون دائماً موجبة ما عدا قيم x التي هي جذور الدالة المعطاة، أى عندما تكون $y(x) = 0$ وبذلك أى قيمة تجعل $y(x)$ لها أقل قيمة سوف تكون أيضاً جذراً للمعادلة $f(x) = 0$.

```

10 REM SEARCH FOR A MAXIMUM OF THE FUNCTION Y=X*COS(X)
20 DEF FNY(X)=X*COS(X)
30 PRINT "LEFT END OF INTERVAL (X1) =";
40 INPUT X1
50 PRINT
60 PRINT "RIGHT END OF INTERVAL (X4) =";
70 INPUT X4
80 PRINT
90 PRINT "MINIMUM SEPARATION BETWEEN INTERIOR POINTS (D) =";
100 INPUT D
110 LET I=1
120
130 REM CALCULATE INTERIOR POINTS
140
150 LET X2=X1+.5*(X4-X1-D)
160 LET X3=X2+D
170 LET Y2=FNY(X2)
180 LET Y3=FNY(X3)
190 PRINT
200 PRINT "Y1=";FNY(X1),"Y2=";Y2,"Y3=";Y3,"Y4=";FNY(X4)
210 PRINT "X1=";X1,"X2=";X2,"X3=";X3,"X4=";X4
220 IF Y2<Y3 THEN 300
230 IF Y2=Y3 THEN 400
240
250 REM Y2 GREATER THAN Y3 - RETAIN LEFT INTERVAL
260
270 LET X4=X3
280 GOTO 340
290
300 REM Y3 GREATER THAN Y2 - RETAIN RIGHT INTERVAL
310
320 LET X1=X2
330
340 REM TEST FOR END OF SEARCH
350
360 IF I=100 THEN 470
370 LET I=I+1
380 IF (X4-X1)>3*D THEN 130
390
400 REM COMPUTE FINAL SOLUTION
410
420 LET X5=.5*(X2+X3)
430 PRINT
440 PRINT "XMAX=";X5,"YMAX=";FNY(X5)
450 STOP
460
470 REM TERMINATE COMPUTATION BECAUSE OF MAXIMUM ITERATION COUNT
480
490 PRINT "MAXIMUM NUMBER OF ITERATIONS EXCEEDED - COMPUTATION ENDS"
500 END

```

شكل ٦ - ٥

٦ - ٣ الدوال المتعددة السطور MULTILINE FUNCTIONS

يوجد عديد من الحسابات التي لا يمكن إجراؤها أو القيام بها في سطر واحد. ويمكن أن يكون ذلك صحيحاً وخصوصاً في الحسابات التي تحتوي على صيغ رياضية طويلة أو على عمليات تفرع مشروط. وتدعم بعض نسخ البيسك صيغة الدالة المتعددة السطور والتي تناسب الحسابات من هذا النوع.

يمكن للدالة المتعددة السطور (مثل الدالة ذات السطر الواحد) ، أن تحتوى على أى عدد من الخلاصات الزائفة كمدخلات ولكنها ترجع قيمة واحدة فقط . يجب أن تكون أول جملة هي جملة DEF . ومع ذلك على عكس الدالة ذات السطر الواحد ، فلا يتضمن تعريف الدالة في جملة DEF . ويجب أن تكون جملة FNEND أى (FUNCTION END) آخر جملة وتكون ببساطة من رقم السطر وتنبه الكلمة الدالة FNEND .

ويمكن أن يوجد بين جملتي DEF و FNEND أى عدد من الجمل والتي تعرف الدالة . ويجب أن تعطى إحدى هذه الجمل قيمة لإسم الدالة . ويتم إنجاز ذلك عادة بجملة LET حيث يظهر إسم الدالة على يسار علامة التساوى (=) . وتستخدم نفس طريقة التسمية التقليدية كى في ادانة ذات السطر الواحد .

```

LEFT END OF INTERVAL (X1) = 70
RIGHT END OF INTERVAL (X4) = 73.14159
MINIMUM SEPARATION BETWEEN INTERIOR POINTS (D) = 7.0001

Y1= 0          Y2= 8.06277E-5          Y3=-7.64780E-5
Y4=-3.14159
X1= 0          X2= 1.57075          X3= 1.57085          X4= 3.14159

Y1= 0          Y2= 0.555356          Y3= 0.555372          Y4=-7.64780E-5
X1= 0          X2= 0.785372          X3= 0.785473          X4= 1.57085

Y1= 0.555356   Y2= 0.450865          Y3= 0.450795          Y4=-7.64780E-5
X1= 0.785372   X2= 1.17806          X3= 1.17816          X4= 1.57085

Y1= 0.555356   Y2= 0.545438          Y3= 0.545412          Y4= 0.450795
X1= 0.785372   X2= 0.981716          X3= 0.981816          X4= 1.17816

Y1= 0.555356   Y2= 0.560534          Y3= 0.560529          Y4= 0.545412
X1= 0.785372   X2= 0.883544          X3= 0.883644          X4= 0.981816

Y1= 0.555356   Y2= 0.560405          Y3= 0.56041          Y4= 0.560529
X1= 0.785372   X2= 0.834458          X3= 0.834558          X4= 0.883644

Y1= 0.560405   Y2= 0.561094          Y3= 0.561095          Y4= 0.560529
X1= 0.834458   X2= 0.859001          X3= 0.859101          X4= 0.883644

Y1= 0.561094   Y2= 0.560972          Y3= 0.560969          Y4= 0.560529
X1= 0.859001   X2= 0.871273          X3= 0.871373          X4= 0.883644

Y1= 0.561094   Y2= 0.561072          Y3= 0.561071          Y4= 0.560969
X1= 0.859001   X2= 0.865137          X3= 0.865237          X4= 0.871373

Y1= 0.561094   Y2= 0.561093          Y3= 0.561093          Y4= 0.561071
X1= 0.859001   X2= 0.862069          X3= 0.862169          X4= 0.865237

Y1= 0.561094   Y2= 0.561096          Y3= 0.561096          Y4= 0.561093
X1= 0.859001   X2= 0.860535          X3= 0.860635          X4= 0.862169

Y1= 0.561094   Y2= 0.561096          Y3= 0.561096          Y4= 0.561096
X1= 0.859001   X2= 0.859768          X3= 0.859868          X4= 0.860635

Y1= 0.561096   Y2= 0.561096          Y3= 0.561096          Y4= 0.561096
X1= 0.859768   X2= 0.860152          X3= 0.860252          X4= 0.860635

Y1= 0.561096   Y2= 0.561096          Y3= 0.561096          Y4= 0.561096
X1= 0.860152   X2= 0.860343          X3= 0.860443          X4= 0.860635

XMAX= 0.860393          YMAX= 0.561096

```

شكل ٦ - ٦

مثال ٦ - ٦

موضح فيما يلى التخطيط الهيكلى للدالة المتعددة السطور :

```

200 DEF FNA(X,Y,Z)
...
250 LET FNA=...
260 FNEND

```


تسمى هذه الدالة FNA وتستخدم خلاصات زائفة X و Y و Z كدخلات . وتعطى قيمة الدالة عند حسابها في الجملة رقم 250 .
القواعد اللغوية التي تطبق على الدوال المتعددة السطور هي نفسها التي تطبق على الدوال ذات السطر الواحد ؛ (فثلاً يمكن أن يظهر تعريف الدالة في أى مكان من البرنامج ، ونشير إلى الدالة بتحديد اسمها تتبعه قائمة من الخلاصات محصورة بين قوسين وتفصل بينها بواسطة فصلة (،) و . . . الخ) وبالإضافة إلى ذلك ، فلا يمكن تحويل التحكم بين جملة بداخل الدالة ونقطة خارج نطاق الدالة .

مثال ٦ - ٨

مبين فيما يلي جزء من برنامج ببسك يحتوى على تعريف دالة متعددة السطور إلى هذه الدالة . الغرض من هذه الدالة هو تحديد أقل قيمة لأزواج من الأعداد .

```
20 DEF FNM(A,B)
30 LET FNM=A
40 IF A<=B THEN 60
50 LET FNM=B
60 FNEND
...
150 PRINT FNM(FNM(C1,C2),FNM(C2,C3))
```

لاحظ تداخل الدالة FNM مع نفسها في رقم الجملة 150 . وتسبب هذه الجملة إيجاد أصغر قيمة من ثلاث كيات مثلة بالمتغيرات C1 و C2 و C3 ثم طباعتها .

يمكن ظهور متغيرات أخرى غير المعطاة كخلاصات في الدالة المتعددة السطور تماماً كما في الدالة ذات السطر الواحد . ويمكن أن تكون هذه المتغيرات إما متغيرات ذات أدلة أو متغيرات بدون أدلة . وتستخدم القيمة الحالية لهذه المتغيرات عند تنفيذ الدالة كل مرة .

مثال ٦ - ٩

برنامج ببسك يحتوى على تعريف الدالة متعددة السطور التالية :

```
100 DEF FNY(X)
110 IF X>300 THEN 140
120 LET FNY=A+B*X+C*X^2
130 GO TO 150
140 LET FNY=D+E*X+F*X^2
150 FNEND
```

لاحظ أن قيمة X تعتمد من خلال خلاصة وذلك عند الإشارة إلى الدالة ، بينما قيم A و B و C و D و E و F لا تعتمد كخلاصات . وبذلك فإن أحدث قيم هذه المتغيرات سوف تستخدم في كل مرة تحسب فيها الدالة .

يجب أن يكون مفهوماً أن الدالة التي تحتوى على خلاصات من نوع معين يمكن أن تنتج قيمة من نوع مختلف (مثال ، الدالة التي لها خلاصات حرفية يمكن أن تستخدم لتحديد قيمة رقمية) . وعلاوة على ذلك ، لا يستلزم أن تكون الخلاصات من نفس النوع (أى يمكن وجود مزيج من خلاصات رقمية وحرفية) . وهذا صحيح لكل من الدالة ذات السطر الواحد والدالة المتعددة الأسطر ومع ذلك تذكر ، أن الخلاصات في الإشارة للدالة يجب أن تناظر الخلاصات في تعريف الدالة من حيث العدد والنوع .

مثال ٦ - ١٠

يمثل الميكل التخطي التالى دالة متعددة الأسطر تتطلب خلاصتين إحداهما رقمية والأخرى حرفية . والدالة نفسها تعطى نتيجة حرفية (لوجود علامة الدولار في إسمها) .

```

100 DEF FNW$(C,N$)
...
140 LET FNW$=...
150 FNEND

```

عند الإشارة إلى هذه الدالة يجب أن نمدّها بختلاصتين إحداهما رقمية والأخرى حرفية ، ويجب أن تكونا بهذا الترتيب . لا يستلزم أن تكون أسماء الخلاصات بالطبع نفس أسماء الخلاصات الزائفة . من ثم فيمكن أن يكون مرجع الدالة المناسب مثل :

```
250 LET N$=FNW$(X,T$)
```

توضح الأمثلة ٦ - ١٥ ، ٦ - ٢٠ استخدام الدوال المتعددة الأسطر في برامج بيسك كاملة .

٦ - ٤ : تكويد وفك شفرة البيانات — جملة CHANGE

ENCODING AND DECODING DATA—THE CHANGE STATEMENT

عندما تمثل سلسلة حرفية للحاسب ، فإن الحروف التي تكون هذه السلسلة لا تخزن في الحاسب كحروف ولكنها تخزن كأرقام مكدودة بتسلسل كل رقم أو حرف أو حرف خاص يمكن تمثيله برقم فريد خاص به .

توجد عدة أنظمة تكويد رقمية مختلفة مستخدمة في الأجهزة المتنوعة . والنظام الأكثر شيوعاً هو نظام ASCII ذو الأرقام السبعة الثنائية (+) ، حيث يمثل الحرف A بالرقم (العشري) 65 . ويمثل الحرف B بالرقم 66 ، ... وهكذا . يبين جدول ٦ - ١ كود ASCII المكافئ لأكثر الحروف شيوعاً .

يتم القيام بعملية التحويل من حروف إلى أرقام ، وبالعكس ، أوتوماتيكياً بداخل الحاسب حتى أن المبرمج لا يشعر حقيقة هذا التحويل ولا حتى حدوثه . ولكن ، في بعض الأحوال يكون المطلوب هو العمل بالرقم المكافئ للحروف في السلسلة الحرفية ، ويسمح ذلك بالتعامل مع كل حرف على حدة . وجملة CHANGE تسمح بالقيام بعمل هذا التحويل .

توجد طريقتان مختلفتان يمكن أن تكتب بهما جملة CHANGE . تتكون الأولى من رقم السطر ثم الكلمة الدالة CHANGE ثم متغير حرفي ثم الكلمة الدالة TO ثم قائمة رقمية ويجب أن تكون بهذا الترتيب . وتسبب هذه الجملة تحويل كل حرف من السلسلة الحرفية إلى الرقم المكافئ له ويخزن في القائمة الرقمية . سوف يشير أول عنصر من القائمة (ذو الدليل صفر) إلى عدد الحروف المكدودة والتي تحتويها القائمة .

مثال ٦ - ١١

يحتوي برنامج بيسك على الجمل التالية :

```
100 CHANGE N$ TO N
```

حيث N اسم القائمة الرقمية نفرض أن N\$ ممثلة بالسلسلة MONDAY وأن كود ASCII المبين في جدول ٦ - ١ قابل

(+) كود أمريكي قياسي لتبادل المعلومات ، وهو كود واسع الانتشار يؤيده المعهد القومى الأمريكى للمعايرة .

(American Standard Code for Information Interchange)

جدول ٦ - ١ كود ASCII ذو الأرقام السبعة الثنائية

كود	حرف	كود	حرف
48	0	65	A
49	1	66	B
50	2	67	C
51	3	68	D
52	4	69	E
53	5	70	F
54	6	71	G
55	7	72	H
56	8	73	I
57	9	74	J
43	+	75	K
45	-	76	L
47	/	77	M
42	*	78	N
94	↑	79	O
40	(80	P
41)	81	Q
60	<	82	R
62	>	83	S
61	=	84	T
63	?	85	U
36	\$	86	V
34	"	87	W
44	,	88	X
46	.	89	Y
59	;	90	Z
13	عودة العربية		
10	تنفيذ أسطر		
32	حرف خال		

للاستعمال . وسوف يسبب تنفيذ جملة CHANGE إعطاء القيم التالية لعناصر القائمة N .

(يشير إلى أن عدد الحروف في السلسلة = ٦)	$N(0) = 6$
(القيمة العددية المكافئة للحرف M)	$N(1) = 77$
(القيمة العددية المكافئة للحرف O)	$N(2) = 79$
(القيمة العددية المكافئة للحرف N)	$N(3) = 78$
(القيمة العددية المكافئة للحرف D)	$N(4) = 68$
(القيمة العددية المكافئة للحرف A)	$N(5) = 65$
(القيمة العددية المكافئة للحرف Y)	$N(6) = 89$

والآن فن الممكن التوصل للقيمة العددية المكافئة لأي حرف من السلسلة بسهولة وذلك بالرجوع إلى الدليل المناسب .

يمكن تبادل موضع المتغير الحرفي والقائمة الرقية في جملة CHANGE . أي أنه يمكن كتابة الجملة كرقم جملة تتبعها الكلمة الدالة CHANGE ثم قائمة رقية ثم الكلمة الدالة TO ثم متغير حرفي . والجملة بهذا الشكل تحول عنصراً القائمة الرقية إلى سلسلة حرفية . التحويل إلى حروف يبدأ من العنصر الثاني للقائمة (أي من العنصر ذي الدليل الذي قيمته 1) . وسوف يشير العنصر الأول من القائمة (ذو الدليل صفر) كما سبق إلى عدد الحروف الموجودة في السلسلة .

مثال ٦ - ١٢

برنامج ببسك يحتوي على الجملة

225 CHANGE L TO A\$

حيث L اسم قائمة رقية . نفرض أن عناصر القائمة L لها القيم التالية والتي تمثل حروف ASCII ذات الأرقام السبعة الثنائية :

L(0)=11	L(6)=32
L(1)=83	L(7)=67
L(2)=65	L(8)=76
L(3)=78	L(9)=65
L(4)=84	L(10)=85
L(5)=65	L(11)=83

عند تنفيذ جملة CHANGE فسوف يتم تحويل عناصر L من L(1) مبتدئاً من العنصر L(1) إلى الحرف المقابل والسلسلة الناتجة من ذلك سوف تعطى للمتغير A\$. ومن ثم فإن المتغير A\$ سوف يمثل بالقيمة SANTA CLAUS

في المثال ٦ - ١٥ سوف نرى برنامج ببسك كاملاً يستخدم جملة CHANGE .

٦ - ٥ دالتي ASC و CHR\$ THE ASC AND CHR\$ FUNCTIONS

تزامن إلى حد بعيد جملة CHANGE الداليتين المكتبتين ASC و CHR\$ الأولى وهي ASC ، تحول أي حرف إلى ما يكافئه من الكود ASCII وبذلك فإن الدالة تقبل حرفاً واحداً كخلاصة .

مثال ٦ - ١٣

اعتبر الجملة :

50 LET C=ASC(P)

وسوف يترتب على هذه الجملة إعطاء القيمة 80 للمتغير C ، حيث 80 في الكود ASCII تكافئ الحرف P .

وبالمثل الجملة :

70 IF L(I)=ASC() THEN 110

سوف يترتب عليها أن يتحول التحكم إلى رقم الجملة 110 إذا كان المتغير ذو الدليل L(I) له القيمة 32 ، حيث الرقم 32 يكافئ كود ASCII للمسافة الحالية .

الفرض من الدالة المكتبية CHR\$ هو عكس الدالة ASC أي أن CHR\$ تستخدم لتحويل قيمة كود ASCII المناظر لحرف إلى الحرف نفسه في هذه الحالة فإن قيمة الخلاصة يجب أن تكون كية رقية معروفة لكود ASCII ، والقيمة غير الصحيحة سيتم بثها أوتوماتيكياً .

اعتبر الجملة :

```
75 LET A$=CHR$(X)
```

إذا كانت القيمة X هي 42 ، فإن A\$ سوف تمثل الحرف ' ' ، حيث 42 هو تمثيل كود ASCII للحرف (' ') .

وبالمثل فإن الجملة :

```
310 PRINT CHR$(L(I))
```

سوف يترتب عليها طباعة الحرف P إذا كانت L(I) لها القيمة 80 . دالتي ASCII و CHR\$ و جملة CHANGE ، متضمنة في المثال التالي .

مثال ٦ - ١٥ مولد بيجلاتين A Piglatin Generator

بيجلاتين صيغة مكودة بالإنجليزية وتستخدم غالباً كلمة بواسطة الأطفال . كلمة بيجلاتين تتكون من كلمة إنجليزية بتحويل الصوت الأول (وغالباً ما يكون الحرف الأول) إلى نهاية الكلمة ثم بعد ذلك إضافة الحرف "a" وبذلك فإن الكلمة cat تصبح "atca" و "BASIC" تصبح "ASICBA" و piglatin تصبح "iglatinpa" و "igpa atinla" إذا تم هجاؤها كلمتين منفصلتين (وهكذا . . .)

ونرغب في كتابة برنامج بييسك يقبل سطرًا من اللغة الإنجليزية ثم يطبع هذا السطر ثانياً مستخدماً طريقة بيجلاتين (Piglatin) .

طريقة إجراء الحسابات Computational Procedure

سوف نفترض أن كل رسالة نصية يمكن طباعتها على سطر واحد يحتوي على 72 حرفاً كما هو متبع على الوحدات الطرفية الخاصة بالمشاركة الزمنية ، مع ترك مسافة خالية بين كل كلمتين متتاليتين . وسوف يكون إجراء الحسابات بعد ذلك مباشراً ويتكون من وسيلة لاستخلاص كل كلمة من الرسالة ، ثم إعادة تنظيم الكلمة ثم إضافة الحرف "a" ثم طباعة الكلمة بعد إعادة تنظيمها . من السهل ، من حيث المبدأ استخلاص كل كلمة من السطر ، حيث تتميز كل كلمة بوجود مكان خال بينها وبين الكلمات الأخرى .

الحسابات التفصيلية خداعة إلى حد ما ، إلا أن مناولة الحروف في السطر بطريقة فردية (أى كل حرف على حدة) فقط يمكن إذا حولنا الحروف أولاً إلى مكافئاتها الرقمية حسب كود ASCII . وقبل طباعة كل من الكلمات التي أعيد تنظيمها يلزم تغيير الأرقام بكود ASCII مرة أخرى إلى حروف . وسوف تستعمل كل من جملة CHANGE ودالة CHR\$ لهذا الغرض .

واستخلاص كيات ASCII التي تمثل كلمة واحدة من القائمة الكاملة لأرقام ASCII يجب أن تتم بكل حذر . سوف نستخدم مؤشرين (متغيرات رقمية) لهذا الغرض . وسوف يشير المؤشر الأول إلى موضع ما يكافئ الحرف الأول من الكلمة بالكود ASCII والمؤشر الثاني سوف يشير إلى أين يوجد الحرف الأخير من الكلمة بكود ASCII وسوف يعاد تحديد قيم المؤشرين بعد إعادة تنظيم كل كلمة .

وسوف نستخدم دالة متعددة الأسطر من أجل استخلاص كل كلمة ، أى لتحديد مواضع المؤشرات . ولا توجد أى مشكلة في تحديد موضع المؤشر الأول ، حيث نعلم أن الكلمة الأولى سوف تبدأ من المكان رقم 1 ، وسوف تبدأ كل كلمة تالية وراء نطاق نهاية الكلمة السابقة بموضعين . بينما لتحديد موضع المؤشر الثاني سوف نفحص كل حرف مكود بكود ASCII وراء نطاق المؤشر الأول حتى نجد مسافة خالية . ومن ثم يتحدد موضع المؤشر الثاني بمكان واحد قبل المسافة الحالية .

المخطط التمهيدي للبرنامج The Program Outline

من أجل كتابة مخطط تمهيدي مفصل لإجراء الحسابات دعنا نعرف أولاً الرموز التالية :

$N\$$ = السطر المعطى من النص (سلسلة حرفية) .

L = قائمة رقمية تحتوى مكافئات كود ASCII للحروف في سطر النص (لاحظ أن L سوف تتكون من 72 كية فردية مكدودة) .

$P1$ = مكان أول حرف من كلمة معينة في قائمة المكافئات بكود ASCII المخزنة في L (رقم ما بين 1 إلى 72) .

$P2$ = مكان آخر حرف من كلمة معينة في قائمة المكافئات بكود ASCII المخزنة في L (رقم ما بين 1 إلى 72) .

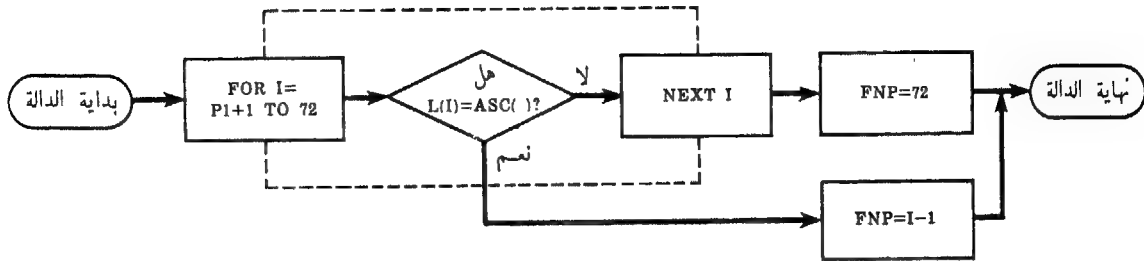
ادرس الدالة متعددة الأسطر FNP والتي ترجع قيمة للمتغير $P2$ عند إعطائها قيمة للمتغير $P1$. سوف تستمر الحسابات كما يلي :

١ - لكل قيمة من قيم الدليل I مبتدئاً من $I = P1 + 1$ ، اختبر لترى إذا كان المتغير ذو الدليل $L(I)$ يحتوى على مكافئ ASCII للمكان الخالي .

(أ) إذا كانت بعض $L(I)$ تمثل مكاناً خالياً (وذلك مشيراً لنهاية الكلمة) ، اجعل $P2 = I - 1$.

(ب) إذا كانت كل $L(I)$ تمثل أى حرف ما عدا المكان الخالي ، اجعل $P2 = 72$ (نهاية السطر) .

٢ - ارجع قيمة $P2$ إلى نقطة الإشارة للدالة . شكل ٦ - ٧ يبين خريطة سير عمليات للإجراء الموصوف عالياً .



شكل ٦ - ٧

سوف نواصل بقية البرنامج على النهج التالي :

١ - اقرأ $N\$$

٢ - اختبر لترى ما إذا كانت $N\$$ تساوى الكلمة END . إذا كانت كذلك أنهى الحسابات ، وإلا واصل بالخطوة 3 التالية .

٣ - حدد قيمة كل عنصر من القائمة L بما يكافئ المكان الخالي من الكود ASCII (وبذلك « تمحى » ما يحتمل أن يكون قد تم تخزينه في L من قبل) .

٤ - حول $N\$$ إلى L

٥ - اجعل $P1 = 1$

٦ - ارجع للدالة FNP لتكوين قيمة $P2$.

٧ - أعد تنظيم الحروف الموجودة في الكلمة ثم اطبعها بالطريقة التالية :

(أ) إذا كانت الكلمة تحتوي فقط على حرف واحد ، فإن P2 سوف تنطبق مع P1 . من ثم واصل مباشرة من الخطوة (٧ ج) أدناه .

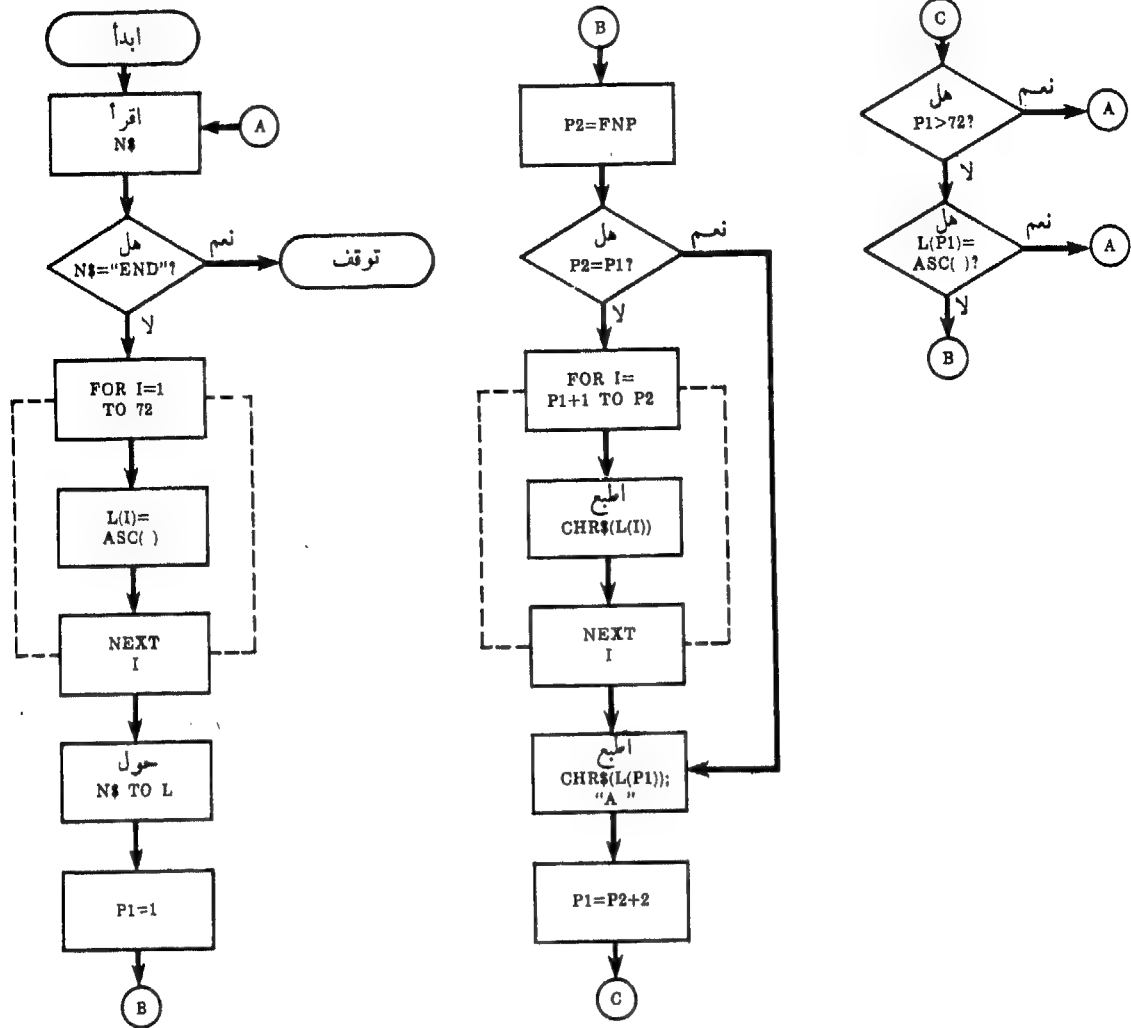
(ب) اطبع الحروف المثلثة بواسطة L(P + 1) إلى L(P2) على التوالي .

(ج) اطبع الحروف المثلثة بواسطة L(P1) ، ويتبعها فوراً الحرف "x" ثم مكان خال .

٨ - كون بداية الكلمة التالية كما يلي :

(أ) اجعل $P1 = P2 + 2$

(ب) إذا تعدت قيمة P1 الرقم 72 ، فإن سطر النص يكون قد اكتمل ، من ثم ارجع مرة أخرى للخطوة 1 واقرأ سطرًا جديدًا من النص .



شكل ٦ - ٨

(ج) إذا كانت $L(PI)$ تمثل سطرًا خالياً (حتى إن لم تكن قيمة PI أكبر من 72) ، وعلى ذلك يكون قد تم الوصول إلى نهاية الرسالة .

(د) اذهب مرة ثانية للخطوة ٦ ويكون قيمة جديدة للمتغير P2 .

لاحظ أن هذه الخطوة سوف تستمر في قراءة سطور متتالية من النص حتى يضاف كلمة END (أنظر خطوات ١ ، ٢ عليه) .

يبين شكل ٦ - ٨ خريطة سير عمليات مناظرة للخطوة سابقة الذكر .

The BASIC Program برنامج بيسك

يحتوي شكل ٦ - ٩ على برنامج البيسك الحقيقي لهذه المسألة. لاحظ استخدام الدالة المكتتبية ASC في السطور 70 و 210 و 390 والدالة CHR\$ في السطور 310 و 330 تستخدم هذه الدوال في تحويل كل حرف من وإلى المكافآت لها بـكود ASCII.

تم برمجة حساب قيمة P2 من قيمة ممطاة P1 كدالة ممتدة الأسطر في السطور 40 إلى 120 ويجب أن نبرز أن هذا البرنامج كان يمكن كتابته بكل سهولة وبدون استخدام دالة معرفة بواسطة المبرمج . وفي هذه الحالة ، فإن الغرض من الدالة هو هيكله جزء مستقل نسبياً من خطة الحسابات كمصدر متساو من برنامج مستقل . ومن ثم فإن ميزة الدالة ميزة تنظيمية أكثر منها حسابية . ومع ذلك وفي بعض البرامج فإن وجود الدالة المعرفة بواسطة المبرمج سوف تخلصه من تكرار برمجة نفس مجموعة الحسابات .

```

10 REM          PIGLATIN GENERATOR
20 DIM L(72)
30
40 DEF FNP(P1)
50 REM          THIS FUNCTION FINDS THE END OF A SINGLE WORD
60 FOR I=P1+1 TO 72
70   IF L(I)=ASC( ) THEN 110
80 NEXT I
90 LET FNP = 72
100 GOTO 120
110 LET FNP=I-1
120 FNEED
130
140 REM          READ A LINE OF TEXT
150
160 PRINT
170 PRINT
180 INPUT N$
190 IF N$="END" THEN 410
200 FOR I=1 TO 72
210   LET L(I)=ASC( )
220 NEXT I
230 CHANGE N$ TO L
240 LET P1=1
250 LET P2=FNP(P1)
260
270 REM          PRINT WORD IN PIGLATIN
280
290 IF P2=P1 THEN 330
300 FOR I=P1+1 TO P2
310   PRINT CHR$(L(I));
320 NEXT I
330 PRINT CHR$(L(P1));"A ";
340
350 REM          FIND NEXT WORD
360
370 LET P1=P2+2
380 IF P1>72 THEN 140          END OF LINE
390 IF L(P1)=ASC( ) THEN 140   END OF TEXT
400 GOTO 250
410 END

```


سوف ترى في شكل ٦ - ١٠ المخرجات المولدة بواسطة البرنامج لثلاثة أسطر نموذجية ، كل سطر يمثل رسالة إدخال يتبناه فوراً السطر المناظر من ييجلاتين . وسوف يتوقف تنفيذ البرنامج بعد إدخال الكلمة END من النهاية الطرفية (لاحظ أن اجابات المستخدم موضوع تحتها خط) .

?THIS IS A PIGLATIN GENERATOR
HISTA SIA AA IGLATINPA ENERATORGA

?WHAT SORT OF GARBLED MESSAGE IS THIS ANYHOW
HATWA ORTSA FOA ARBLEDGA ESSAGEMA SIA HISTA NYHOWAA

?NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY
OWNA SIA HETA IMETA ORFA LLAA OODGA ENMA OTA OMECA OTA HETA IDAA FOA HEI
RTA OUNTRYCA

?END

شكل ٦ - ١٠

٦ - ٦ توليد أرقام عشوائية — دالة RND

GENERATING RANDOM NUMBERS—THE RND FUNCTION

هناك عدة تطبيقات شائعة للحاسب مبنية على أساس توليد أرقام عشوائية . في اليبسك يكون من اليسير جداً توليد الأرقام العشوائية بواسطة الدالة المكتنية RND . وترجع الدالة أرقاماً عشوائية مختلفة بقيمة تتراوح ما بين الصفر والواحد الصحيح ، في كل مرة يرجع فيها للدالة . ولا تتطلب أى خلاصة .

مثال ٦ - ١٦

```
20 DIM X(100)
...
50 FOR I=1 TO 100
60   LET X(I)=RND
70 NEXT I
```

يحتوي برنامج يبسك الجمل التالية :

سوف تتسبب هذه الجملة في توليد 100 رقم عشوائى وتخزينها في قائمة X . كل رقم عشوائى سوف يكون كية عشرية قيمتها تقع ما بين الصفر والواحد الصحيح .

مثال ٦ - ١٧

نفرض أننا نرغب في توليد أرقام عشوائية تتراوح قيمها ما بين 3 و 7 يمكن إنجاز ذلك بكتابة :

```
100 LET X=3+(7-3)*RND
```

أو ببساطة

```
100 LET X=3+4*RND
```

مثال ٦ - ١٨

الجملة المبينة فيما بعد سوف تولد رقماً عشوائياً قيمته صحيحة وتقع ما بين 1 إلى 6 سوف تحدث جميع الأرقام بأرجحية متساوية .

```
100 LET X=1+INT(6*RND)
```

وبتفسير هذه الجملة . يجب أن يكون مفهوماً أن دالة RND يمكن أن ترجع قيمة قريبة جداً من الواحد ولكن ليست واحداً . وبذلك

فإن RND ترجع قيمة 0.99999999 ثم $6 * RND$ سوف تنتج قيمة 5.9999994 و $INT(6 * RND)$ سوف تنتج قيمة 5 . حيث تحدد قيمة X إذن بالرقم 6 .

سوف نرى برنامج ببسك كاملا يستخدم دالة RND في مثال ٦ - ٢٠ .

٦ - ٧ جملة RANDOMIZE THE RANDOMIZE STATEMENT

الأرقام التي نحصل عليها من دالة RND ليست عشوائية تماماً ، حيث أنها تولد باستخدام إجراء حسابات منتظمة . ومع ذلك ، تظهر مثل هذه الأرقام كأنها عشوائية ولها نفس الخواص الإحصائية كأرقام العشوائية الحقيقية . ولذلك فداًئماً ما يشار إلى هذه الأرقام بالأرقام العشوائية الكاذبة .

وفي كل مرة ينفذ فيها برنامج يحتوى على دالة RND سوف تولد نفس الأرقام العشوائية - الكاذبة وب نفس التسلسل . هذا التوالد للأرقام العشوائية يكون عاملاً مساعداً عند تتبع البرنامج لاكتشاف الخطأ . ومن الناحية الأخرى ، فغالباً ما يكون المطلوب هو توليد تسلسل مختلف من الأرقام العشوائية الكاذبة في كل مرة ينفذ فيها برنامج التتبع . ويمكن إنجاز ذلك بواسطة جملة RANDOMIZE .

تتكون جملة RANDOMIZE ببساطة من رقم السطر تتبته الكلمة الدالة RANDOMIZE . والفرض منها هو إعطاء نقطة بداية مختلفة لمولد الأرقام العشوائية . وبذلك فيجب أن تسبق جملة RANDOMIZE المرجع الأول للدالة RND في البرنامج .

مثال ٦ - ١٩

برنامج ببسك يحتوى على الجمل :

```
20 DIM X(100)
30 RANDOMIZE
...
50 FOR I=1 TO 100
60   LET X(I)=RND
70 NEXT I
```

سوف تسبب هذه الجمل في توليد 100 رقم عشوائى كاذب وتخزينها في القائمة X ، كما في المثال ٦ - ١٦ . بينما يختلف عن مثال ٦ - ١٦ ، في مجموعة من الأرقام العشوائية المختلفة سوف يتم توليدها في كل مرة ينفذ فيها البرنامج .

مثال ٦ - ٢٠ لعبة حظ (تصويب كرايس) A Game of Chance (Shooting Craps)

في هذا المثال سوف نحاكى لعبة كرايس على الحاسب . والكرايس لعبة زهر شعبية حيث يرمى اللاعب زوجاً من النرد (الزهر) مرة أو أكثر حتى يكسب أو يخسر . ويمكن تنفيذ اللعبة على الحاسب بالتعويض بالأرقام العشوائية المولدة عن الرمي الفعل للزهر .

قواعد اللعبة Rules of the Game

توجد طريقتان يمكن بهما أن يكسب اللاعب في لعبة الكرايس . يمكنه رمي الزهر مرة ويحصل فيها على 7 أو 11 نقطة أو يمكن أن يحصل على 4 أو 5 أو 6 أو 8 أو 9 أو 10 من الرمية الأولى ، ثم يحصل على نفس النقاط في رميات متعاقبة قبل الحصول على 7 نقاط . وعلى النقيض فهناك طريقتان يمكن أن يخسر بهما اللاعب يمكنه إما أن يرمى الزهر مرة ويحصل على نقطتين أو 3 أو 12 أو يحصل على 4 نقاط أو 5 أو 6 أو 8 أو 9 أو 10 في الرمية الأولى ، ثم يحصل على 7 نقاط في الرمية التالية قبل الرجوع بنفس عدد النقاط التي حصل عليها في الرمية الأولى .

طريقة إجراء الحسابات Computational Procedure

دعنا نبرمج اللعبة بطريقة تفاعلية ، حيث تحاكي رمية الزهر في كل مرة يضغط اللاعب لرجوع العربة على النهاية الطرفية . بعد ذلك تظهر رسالة مشيرة إلى نتيجة كل رمية . وسوف تستمر اللعبة حتى يطبع اللاعب كلمة END وبالإضافة إلى ذلك ، فسوف نصنع طباعة قواعد اللعبة إذا ما أدخل اللاعب الكلمة RULES .

من أجل محاكاة رمية واحدة للزهر سوف تولد رقمين عشوائيين كل منهما يحتوي على رقم صحيح يقع ما بين 1 إلى 6 مجموع هذين العددين سوف يمثل الهدف الذي تم الحصول عليه عند رمية الزهر . ومن المناسب أن تستخدم دالة معرفة بواسطة المبرمج لهذا الغرض حيث أن كل مرة يتم فيها الإشارة إلى الدالة سوف تتم محاكاة رمية أخرى للزهر .

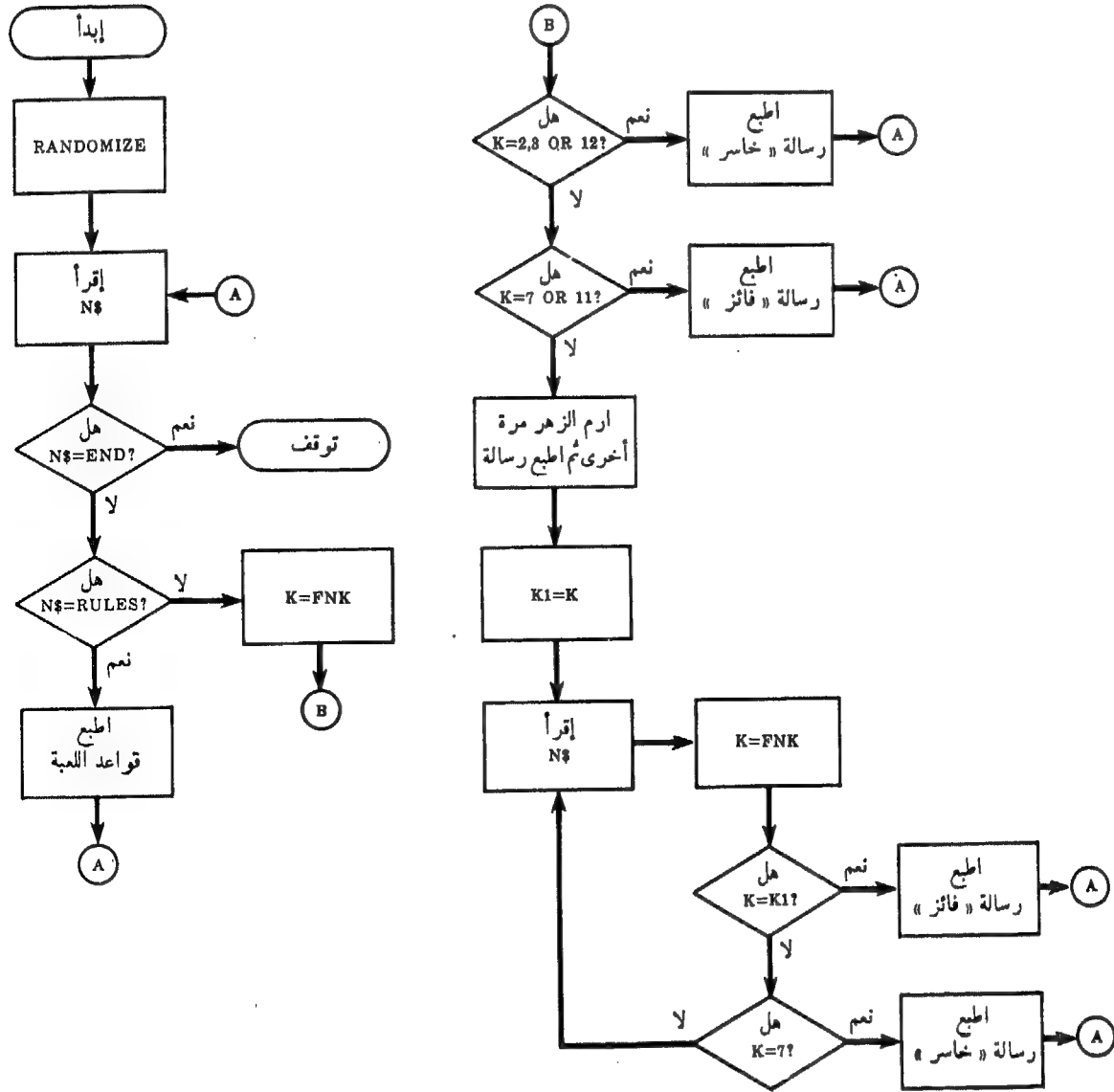
الجزء الرئيسي من البرنامج سوف يفحص الهدف الذي تم الحصول عليه من كل رمية ويحدد ما إذا كان اللاعب من الخاسرين أو الناجحين أو يتطلب رمية أخرى . وفي كل حالة سوف تتم طباعة رسالة ملائمة مع الهدف الذي تمت محاكاته ، وسوف يتضمن أيضاً كتلة من جمل PRINT والتي يترتب عليها طباعة القواعد للإجابة على اللاعب الذي يكتب RULES .

المخطط التمهيدى للبرنامج The Program Outline

- ١ - استهل مولد الرقم العشوائى .
- ٢ - اقرأ قيمة للمتغير الحرفى N\$.
(أ) أوقف التنفيذ إذا كانت N\$=END
(ب) اطبع قواعد اللعبة إذا كانت N\$=RULES ثم كرر هذه الخطوة (أى ، اقرأ قيمة جديدة للمتغير N\$)
(ج) تقدم إلى الخطوة ٣ إذا كانت N\$ تمثل سلاسل حرفية غير END أو RULES . (لاحظ أن أى حرف على حدة يمكن أن يعطى للمتغير N\$ لهذا الغرض . يمكن أن يكون أكثرها ملائمة لهذا الغرض قضيب ترجيع العربة ، وذلك لوضوحه على الآلة الكاتبة الطرفية) .
- ٣ - قم بمحاكاة رمى الزهر مرة واحدة ، مستخدماً نتيجة الهدف (K) .
(أ) إذا كانت قيمة K هي 2 أو 3 أو 12 فذلك يشير إلى الخسارة ثم بعد ذلك اطبع رسالة ملائمة وارجع إلى الخطوة ٢ مرة أخرى .
(ب) إذا كانت قيمة K هي 7 أو 11 فذلك يشير إلى المكسب . ثم بعد ذلك اطبع رسالة ملائمة وارجع إلى الخطوة ٢ مرة أخرى .
(ج) إذا كانت K قيمتها 4 أو 5 أو 6 أو 8 أو 9 أو 10 فإن ذلك يتطلب رمية إضافية للزهر . ثم اطبع رسالة ملائمة وتقدم إلى الخطوة رقم 4 .
- ٤ - اجعل $K1 = K$ (سوف يسمح بمقارنة القيم المتعاقبة للمتغير K أن تقارن بالقيمة الأصلية ، حيث تسمى الآن K1) .
- ٥ - اقرأ قيمة جديدة للمتغير N\$ ثم قم بمحاكاة رمية أخرى للزهر ، وبذلك تولد قيمة جديدة للمتغير K (لاحظ أن الزهر لن يرمى حتى يعطى اللاعب إشارة بإدخال بعض الحروف ، ومن أمثال هذه الحروف إرجاع العربة من أجل N\$) .
- ٦ - قارن K بالمتغير K1 .
(أ) إذا كانت $K1 = K$ فإن ذلك يشير إلى المكسب . من ثم اطبع رسالة ملائمة ثم ارجع للخطوة رقم ٢ .
(ب) إذا كانت قيمة $K = 7$ فإن ذلك يشير إلى المكسب . إذن اطبع رسالة ملائمة ثم ارجع للخطوة رقم ٢ .

(٢) إذا كانت K لا تساوي $K1$ ولا تساوي 7 ، إذن ارجع للخطوة رقم ٥ وقم بتوليد قيمة جديدة للمتغير K .

خريطة لسير عمليات هذا الإجراء المبينة في شكل ٦ - ١١ .



شكل ٦ - ١١

سوف نقوم بعملية محاكاة رمية واحدة للزهر في دالة معرفة بواسطة المبرمج كما يلي :

$$K2 = 1 + \text{INT}(6 \cdot \text{RND}) \quad \text{١ - اجمل}$$

(أنظر المثال ٦ - ١٨ لشرح هذه الجملة)

$$K3 = 1 + \text{INT}(6 \cdot \text{RND}) \quad \text{٢ - اجمل}$$

$$K = K2 + K3 \quad \text{٣ - اجمل}$$

٤ - ارجع القيمة الحالية للمتغير K إلى نقطة الإشارة للدالة .

يبين شكل ٦ - ١٢ خريطة سير عمليات مناظرة له . لاحظ أن القيمة المحسوبة للمتغير K تسمى FNK بداخل الدالة .



شكل ٦ - ١٢

برنامج بيسك The BASIC Program

يحتوى شكل ٦ - ١٣ على برنامج كامل للقيام بالحسابات . لاحظ أن السطور من 500 إلى 560 تعرف دالة متعددة السطور FNK حيث تحاكي رمية زوج من الزهر . وقد تم الرجوع إلى هذه الدالة مرتين في نقطتين مختلفتين في البرنامج - بالتحديد السطور 90 و 180 . سوف نرى من خلال الدالة FNK المعرفة بواسطة المبرمج أننا أشرنا إلى الدالة المكتوبة RND ولاحظ، أيضاً جملة RANDOMIZE في السطر رقم 20 ، والذي استخدم ليستهل مولد الرقم العشوائى في كل مرة ينفذ فيها البرنامج . من المهم أن تسبق جملة RANDOMIZE الإشارة إلى الدالة RND (من خلال الإشارة إلى الدالة FNK)

```

10 REM          SIMULATION OF A GAME OF CRAPS
20 RANDOMIZE
30 DEF FNK
40 LET K2=1+INT(6*RND)
50 LET K3=1+INT(6*RND)
60 LET FNK=K2+K3
70 FNEND
80 INPUT N$
90 IF N$="END" THEN 460
100 IF N$="RULES" THEN 330
110
120 REM          SIMULATE ONE PLAY OF CRAPS
130
140 LET K=FNK
150 ON (K-1) GOTO 160,160,200,200,200,180,200,200,200,180,160
160 PRINT K; " - YOU LOSE ON THE FIRST THROW"
170 GOTO 300
180 PRINT K; " - YOU WIN ON THE FIRST THROW"
190 GOTO 300
200 LET K1=K
210 PRINT K; " - THROW THE DICE AGAIN"
220 INPUT N$
230 LET K=FNK
240 IF K=K1 THEN 270
250 IF K=7 THEN 290
260 GOTO 210
270 PRINT K; " - YOU WIN BY MATCHING YOUR FIRST SCORE"
280 GOTO 300
290 PRINT K; " - YOU LOSE BY FAILING TO MATCH YOUR FIRST SCORE"
300 PRINT
310 GOTO 80
320
330 REM          PRINT RULES OF CRAPS
340
350 PRINT,"RULES OF CRAPS"
360 PRINT
370 PRINT "TO WIN: OBTAIN A 7 OR 11 ON THE FIRST THROW"
380 PRINT "      OR: OBTAIN A 4,5,6,8,9 OR 10 ON THE FIRST THROW"
390 PRINT,"AND MATCH YOUR ORIGINAL SCORE BEFORE THROWING A 7"
400 PRINT
410 PRINT "TO LOSE: OBTAIN A 2,3 OR 12 ON THE FIRST THROW"
420 PRINT "      OR: OBTAIN A 4,5,6,8,9 OR 10 ON THE FIRST THROW"
430 PRINT,"AND THROW A 7 BEFORE MATCHING YOUR ORIGINAL SCORE"
440 PRINT
450 GOTO 80
460 END
  
```

مثال ٦ - ٢١

نبين فيما يلي برنامج فرعى نموذجي :

```
300 REM SUBROUTINE TO CALCULATE CRITICAL CONSTANTS
310 LET C1=(A+B·C)/3
320 LET C2=SQR(A↑2+B↑2+C↑2)
330 LET C3=SQR(A·B·C)
340 RETURN
```

لاحظ أن البرنامج الفرعى يبدأ بجملة REM وينتهى بجملة RETURN يجب أن تحدد المتغيرات A و B و C قيما عددية قبل الإشارة إلى البرنامج الفرعى .

يمكن أن يحتوى البرنامج الفرعى الصغير عدة جمل RETURN عند الحاجة وغالباً ما يكون ذلك ضرورياً وله أهمية كبيرة إذا كان البرنامج الفرعى يحتوى على تفرع مشروط أو تفرع متعدد . وعند تنفيذ البرنامج الفرعى فإن أول جملة RETURN يقابلها سوف تسبب تحويل التحكم من البرنامج الفرعى .

مثال ٦ - ٢٢

الميكمل التخطيطي لبرنامج فرعى يحتوى على عدة جمل RETURN مبينة فيما يلي :

```
500 REM SAMPLE SUBROUTINE WITH MULTIPLE RETURNS
510 ON N 520,580,650
520 LET X=...
...
570 RETURN
580 LET Y=...
...
640 RETURN
650 LET Z=...
...
690 RETURN
```

يحتوى هذا البرنامج الفرعى على تفرعات عديدة حيث يتحول التحكم إلى الجمل 520 و 580 و 650 (و كل ذلك يتضمنه البرنامج الفرعى) متوقفاً في ذلك على قيمة N . ثم بعد ذلك يتحول التحكم مرة أخرى إلى الجملة التى تلى الإشارة للبرنامج الفرعى عند مقابلة أى جملة . من جمل RETURN .

٦ - ٩ الإشارة إلى برنامج فرعى — جملة GOSUB

REFERENCING A SUBROUTINE—THE GOSUB STATEMENT

يمكن الإشارة إلى البرنامج الفرعى بواسطة جملة GOSUB تتكون هذه الجملة من رقم للسطر ، ثم الكلمة الدالة GOSUB ثم بعد ذلك رقم أول جملة في البرنامج الفرعى . وعند تنفيذ هذه الجملة فإنها تحول التحكم إلى البرنامج الفرعى . وسوف يتحول التحكم مرة أخرى إلى الجملة التى تلى GOSUB عند مقابلة جملة RETURN بداخل البرنامج الفرعى .

مثال ٦ - ٢٣

```
120 GOSUB 300
130 PRINT "Z=";Z
...
300 LET X=A+B
...
380 RETURN
```

برنامج ببسك يحتوى على الجمل التالية :

برنامج فرعى صغير

في شكل ٦ - ١٤ نرى قائمة تمثل مخرجات البيانات . أولاً نرى أن القواعد قد تمت طباعتها عند طلب اللاعب لذلك بكتابة **RULES** . يلي ذلك ٥ ألعاب نمطية (٣ مكسب و ٢ خسارة) . وأخيراً كتب اللاعب **NED** ، ويترتب على ذلك إنهاء تنفيذ البرنامج . (لاحظ أن استجابته المستخدم تحتها خط .

?RULES**RULES OF CRAPS**

TO WIN: OBTAIN A 7 OR 11 ON THE FIRST THROW
OR: OBTAIN A 4,5,6,8,9 OR 10 ON THE FIRST THROW
AND MATCH YOUR ORIGINAL SCORE BEFORE THROWING A 7

TO LOSE: OBTAIN A 2,3 OR 12 ON THE FIRST THROW
OR: OBTAIN A 4,5,6,8,9 OR 10 ON THE FIRST THROW
AND THROW A 7 BEFORE MATCHING YOUR ORIGINAL SCORE

?
11 - - YOU WIN ON THE FIRST THROW
?
5 - - THROW THE DICE AGAIN
?
7 - - YOU LOSE BY FAILING TO MATCH YOUR FIRST SCORE
?
3 - - YOU LOSE ON THE FIRST THROW
?
7 - - YOU WIN ON THE FIRST THROW
?
4 - - THROW THE DICE AGAIN
?
8 - - THROW THE DICE AGAIN
?
9 - - THROW THE DICE AGAIN
?
11 - - THROW THE DICE AGAIN
?
6 - - THROW THE DICE AGAIN
?
5 - - THROW THE DICE AGAIN
?
4 - - YOU WIN BY MATCHING YOUR FIRST SCORE

?END

شكل ٦ - ١٤

٦ - ٨ تعريف برنامج فرعي DEFINING A SUBROUTINE

في بعض الأحيان يكون من الأكثر ملاءمة بناء هياكل من الجمل المتتالية كبرنامج فرعي بدلاً من دالة . البرامج الفرعية مشابهة للدول بمعنى أننا يمكن الرجوع إليها من أماكن أخرى من البرنامج . يختلف البرنامج الفرعي عن الدالة من حيث أنه ليس له اسم كما أن البرنامج الفرعي يمكن أن يستخدم ليحدد قيماً لأكثر من كمية عددية و / أو سلاسل حرفية . وعلاوة على ذلك لا نستخدم الخلاصات . ومن ثم فإن البرنامج الفرعي يتبادل المعلومات مع باقي البرنامج بطريقة عامة جداً .

لا يستلزم أن نبدأ البرنامج الفرعي بجملة خاصة . وعلى ذلك يمكننا بداية البرنامج الفرعي بجملة **REM** أو جملة **LET** أو جملة **FOR - TO** أو جملة إدخال (الخ) . ولكن يجب أن يكون آخر جملة هي جملة **RETURN** حيث تتكون ببساطة من رقم جملة تتبعه الكلمة الدالة **RETURN** وتسبب هذه الجملة تحويل التحكم مرة أخرى إلى الجملة الموجودة التي تلي نقطة الإشارة إلى البرنامج الفرعي . (يجب أن يفهم أن التحكم لا يمكن أن يتحول بواسطة أي جملة من جمل التفرع ، كمثل **GO TO** أو **IF-THEN** أو **ON-GO TO**) .

عند مقابلة الجملة رقم 120 (GOSUB) أثناء تنفيذ البرنامج . سوف يتحول التحكم إلى الجملة رقم 300 وسوف ينفذ البرنامج الفرعى . وعند الوصول إلى الجملة رقم 380 (RETURN) فسوف يتحول التحكم مرة أخرى إلى الجملة رقم 130 وهى أول جملة بعد GOSUB .

يمكن أن يحتوى البرنامج على أكثر من استدعاء لنفس البرنامج الفرعى . ودائماً يتحول التحكم مرة أخرى من البرنامج الفرعى إلى الجملة التى تلي جملة GOSUB المعنية التى أشارت إلى البرنامج الفرعى .

مثال ٦ - ٢٤

برنامج ببسك يحتوى على الجمل التالية :

```
120 GOSUB 300
130 PRINT "Z=";Z
...
180 GOSUB 300
190 IF Z<10 THEN 250
...
300 LET X=A+B
...
380 RETURN
```

} برنامج فرعى صغير

تعرف الجمل 300 إلى 380 البرنامج الفرعى ، كما فى مثال ٦ - ٢٣ . إذا استدعى البرنامج الفرعى بواسطة الجملة رقم 120 ، فإن التحكم سوف يعود مرة أخرى إلى الجملة رقم 130 بعد إتمام تنفيذ البرنامج الفرعى . وبالمثل ، إذا استدعى البرنامج الفرعى بواسطة الجملة رقم 180 ، فإن التحكم سوف يعود مرة أخرى إلى الجملة رقم 190 بعد إتمام تنفيذ البرنامج الفرعى .

من الممكن أن يستدعى برنامج فرعى برنامجاً فرعياً آخر . والبرامج الفرعية التى تم هيكلتها بهذه الطريقة تسمى برامج فرعية متداخلة (تذكر أننا قابلنا هذا المصطلح فى الفصل الرابع ، عند مناقشة الحلقات التكرارية المتداخلة FOR-TO) .

مثال ٦ - ٢٥

برنامج ببسك يحتوى على الجمل التالية :

```
50 GOSUB 200
...
200 LET C=A+B
...
240 GOSUB 300
...
270 RETURN
...
300 LET P=Q+R
...
350 RETURN
```

} البرنامج الفرعى الأول

} البرنامج الفرعى الثانى

يحتوى هذا البرنامج على برنامجين فرعيين . يتكون البرنامج الفرعى الأول من الجمل 200 حتى 270 ، ويتكون البرنامج الفرعى الثانى من الجمل 300 حتى 350 لاحظ أننا استدعينا البرنامج الفرعى الثانى من نقطة بداخل البرنامج الفرعى الأول (السطر 240) ، ومن ثم فإن البرنامجين الفرعيين متداخلين .

عند مقابلة الجملة رقم 350 أثناء تنفيذ البرنامج ، فإن التحكم سوف يتحول مرة ثانية إلى الجملة التى تلى السطر 240 . وبذلك فإن التحكم تحول من البرنامج الفرعى الثانى إلى البرنامج الفرعى الأول . وبالمثل فإن الجملة رقم 270 سوف تحول التحكم ثانية إلى الجملة التى تلى السطر رقم 50 ، وبذلك يتم تحويل التحكم من البرنامج الفرعى الأول إلى نقطة الاستدعاء الأول .

دعنا نحسب الدخل الإجمالي الشهري المعدل ، وضريبة الولاية والضريبة المحلية في برنامج فرعى ، ثم الضريبة الفيدرالية في برنامج فرعى آخر . ويسمح لنا ذلك بفصل البرنامج إلى عدة أجزاء منفصلة . وسوف تكون البرامج الفرعية متداخلة وذلك باستدعاء البرنامج الفرعى للضريبة الفيدرالية بواسطة البرنامج الفرعى الآخر . أما باقى البرنامج فسوف يقرأ البيانات المطلوبة ببساطة (وهى اسم الموظف ورقه والمرتب الإجمالي الشهري والحالة الاجتماعية وعدد المعولين) ثم يستدعى البرامج الفرعية ، وبعد ذلك يحسب صافى المرتب الشهري وأخيراً يطبع النتائج المطلوبة (وهى : مقدار الضريبة الفيدرالية وضريبة الولاية والضريبة المحلية التى يجب استقطاعها وصافى المرتب الشهري) لكل موظف .

وسوف نشير من داخل البرنامج الفرعى للضريبة الفيدرالية إلى الأرقام الموجودة في أعمدة الطرف الأيسر من جدول الضرائب ،
والذى يقرر حدود الدخل العديدة ، كعناصر للمجموعة المتراسة C وسوف نعرف C على أنها مجموعة متراسة ذات بعدين (6×2)
إذا كان الدليل الثانى لـ يساوى 1 ، سوف تشير إلى بيانات الرجل الأعزب ، بينما عندما $J = 2$ تشير إلى بيانات الرجل المتزوج ومن ثم .
 $C(1,1) = 88$ و $C(2,1) = 133$ و $C(6,1) = 917$ و $C(1,2) = 88$ و $C(2,2) = 183$ و $C(6,2) = 1667$ و ...

وبطريقة مماثلة اجعل T مجموعة متراسة ذات بعدين (6×2) تحتوى على قيمة أساس حدود ضريبة الدخل ، R مجموعة متراسة ذات
بعدين (6×2) تمثل نسبة الضريبة (كرقم عشرى) لكل من حدود الدخل . وبذلك فإن $T(1,1) = 0$ و $T(2,1) = 6.30$ و
 $T(6,1) = 160.92$ و $T(1,2) = 0$ و $T(2,2) = 13.30$ و $T(6,2) = 291.01$ ، ... و $R(1,1) = 0.14$ و $R(2,1) = 0.17$ و $R(6,1) = 0.24$ و ... و $R(1,2) = 0.14$ و $R(2,2) = 0.17$ و $R(6,2) = 0.25$ ، ...

المخطط التمهيدى للبرنامج The Program Outline

دعنا نعرف الرموز التالية :

M\$ = اسم الموظف	P2 = المرتب الإجمالى الشهرى المعدل .
N\$ = الحالة الاجتماعية (M للمتزوج و S للأعزب)	P3 = صافى المرتب الشهرى
N = رقم الموظف	T1 = الضريبة المستقطعة للولاية
E = عدد المعولين	T2 = الضريبة المحلية المستقطعة
P1 = المرتب الأساسى الشهرى	T3 = الضريبة الفيدرالية المستقطعة

وسوف تم العمليات الحسابية كما هو مبين فيما يلى :

١ - حدد قيم رقمية للمجموعات المتراسة C و R و T ذات البعد (6×2) وذلك بواسطة جملتى READ و DATA (حلقتين
تكراريتين من حلقات FOR-TO سوف تكون مطلوبة لكل مجموعة متراسة ، كما ناقشنا ذلك في الفصل الخامس) .

٢ - اقرأ قيمة حرفية مناسبة للمتغير N\$

(أ) إذا كانت N\$=END إنه الحسابات

(ب) وإلا فواصل للخطوة رقم ٣ التالية

٣ - اقرأ قيم رقمية وحرفية لكل من N و P1 و M\$ و E

٤ - احسب قيمة رقمية لكل من T1 و T2 و T3 وذلك باستدعاء البرامج الفرعية المناسبة .

٥ - احسب قيمة P3 باستخدام الصيغة

$$P3 = P1 - (T1 + T2 + T3)$$

٦ - اطبع القيم الحالية لكل من T3 و T1 و T2 و P3 .

٧ - ارجع للخطوة رقم ٢ وأبدأ معالجة بيانات الموظف التالى :

وسوف نواصل البرنامج الفرعى الذى يشار إليه مباشرة في الخطوة رقم ٤ كما يلى :

١ - افحص قيمة P1

(أ) إذا لم تزيد P1 عن \$ 600 ، فاحسب قيمة T1 (ضريبة الولاية) .

$$T1 = 0.01 * P1$$

ثم واصل إلى الخطوة رقم ٢ أدناه .

(ب) إذا زادت قيمة P1 عن \$600 ولكن لم تتعد \$2000 ، فاحسب قيمة T1 باستخدام الصيغة :

$$T1 = 6 + 0.015 * (P1 - 600)$$

ثم واصل إلى الخطوة رقم ٢ أدناه .

(ج) أما إذا زادت قيمة P1 عن \$2000 ، فاحسب T1

$$T1 = 27 + 0.02 * (P1 - 2000)$$

ثم واصل إلى الخطوة رقم ٢ أدناه .

٢ - ومرة أخرى افحص قيمة P1

(أ) إذا لم تزيد قيمة P1 عن \$800 ، احسب قيمة T2 (الضريبة المحلية)

$$T2 = 0.01 * P1$$

ثم واصل إلى الخطوة رقم ٣ أدناه .

(ب) إذا زادت قيمة P1 عن \$800 اجعل قيمة T2 = 8 ثم واصل للخطوة رقم ٣ أدناه .

٣ - احسب قيمة P2 (المرتب الشهري الأساسي المعدل) باستخدام الصيغة

$$P2 = P1 - 54.20 * E$$

(أ) إذا كانت P2 أقل أو تساوى صفراً ، اجعل قيمة T3 (الضريبة الفيدرالية) تساوى صفراً ثم ارجع مرة أخرى إلى الجزء الأساسي من البرنامج .

(ب) أما إذا كانت P2 لها قيمة موجبة إذن ارجع للبرنامج الفرعي الذي يقرر قيمة T3 ثم ارجع مرة أخرى للجزء الأساسي من البرنامج .

وسوف يواصل البرنامج الفرعي الذي يحسب قيمة T3 (الضريبة الفيدرالية) بالطريقة التالية :

١ - قرر ما إذا كان الموظف أعزب أو متزوجاً وذلك بفحص السلسلة الحرفية M\$.

(أ) إذا كانت M\$=S ، اعط قيمة 1 إلى J ثم واصل للخطوة رقم ٢

(ب) إذا كانت M\$=M ، اجعل قيمة 2 إلى J ثم واصل للخطوة رقم ٢ أدناه .

٢ - اجعل قيمة T3 المبدئية مساوية للصفر .

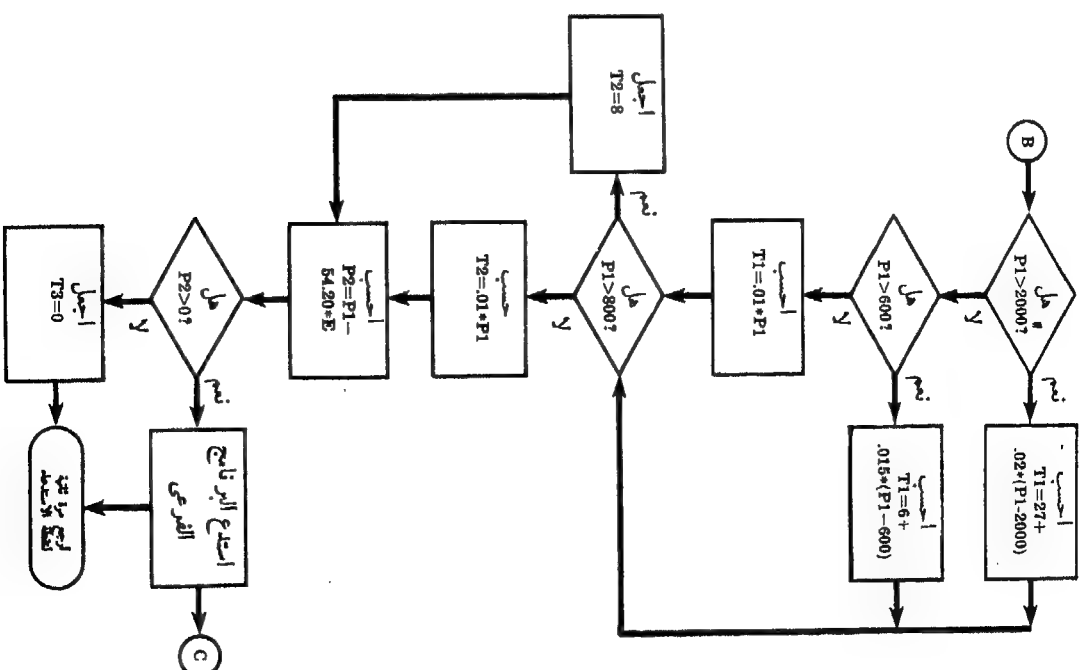
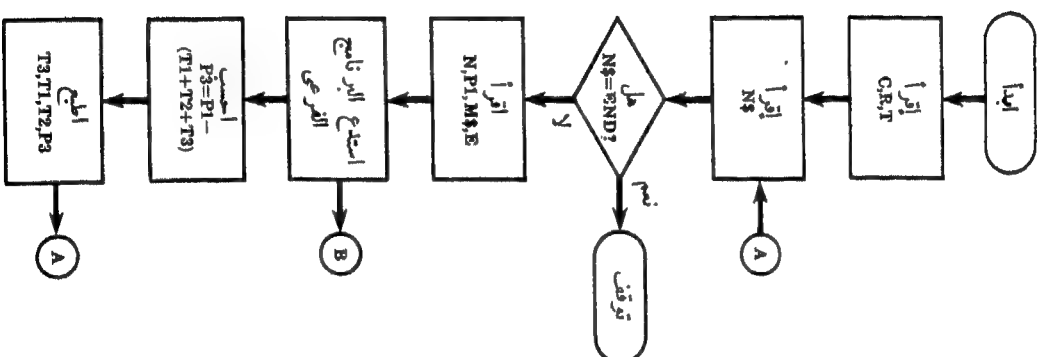
٣ - قارن قيمة P2 بالعنصر C(I, J) لكل قيم I التي تتراوح من 1 إلى 6

(أ) إذا كانت P2 لا تتعدى قيمة C(1, J) احتفظ بقيمة T3 = 0 ثم ارجع إلى نقطة استدعاء البرنامج الفرعي .

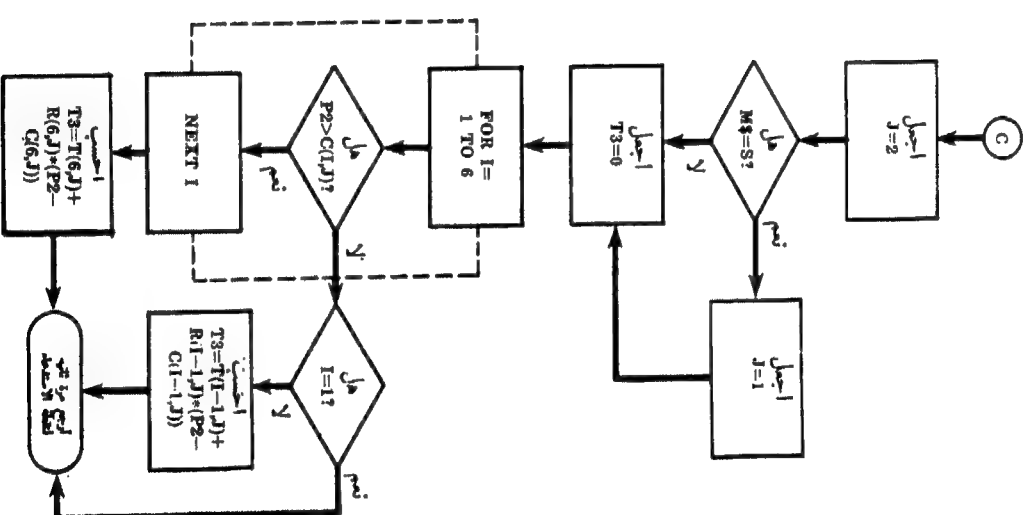
(ب) إذا تعدت قيمة P2 قيمة C(1, J) واصل خلال الحلقة التكرارية حتى نصل إلى قيمة C(I, J) التي تتعدى P2 . إذا وجدت مثل هذه القيمة فاحسب قيمة T3 .

$$T3 = T(I-1, J) + R(I-1, J) * (P2 - C(I-1, J))$$

ثم ارجع إلى نقطة استدعاء البرنامج الفرعي .



شكل ١٥-٦



(=) إذا تعدت قيمة P2 قيمة C(I, J) عند نهاية الحلقة التكرارية (أى عندما تكون I = 6) ، إذن احسب قيمة T3 باستخدام الصيغة

$$T3 = T(6, J) + R(6, J) * (P2 - C(6, J))$$

ثم ارجع مرة أخرى إلى نقطة استدعاء البرنامج الفرعى .

يحتوى شكل ٦ - ١٥ على خريطة سير عمليات مفصلة تناظر المخطط التهيدي السابق .

```

10 REM      COMPUTATION OF A MONTHLY PAYROLL
20 DIM C(6,2),R(6,2),T(6,2)
30 FOR J=1 TO 2
40   FOR I=1 TO 6
50     READ C(I,J)
60   NEXT I
70 NEXT J
80 FOR J=1 TO 2
90   FOR I=1 TO 6
100    READ R(I,J)
110   NEXT I
120 NEXT J
130 FOR J=1 TO 2
140   FOR I=1 TO 6
150    READ T(I,J)
160   NEXT I
170 NEXT J
180 PRINT,"MONTHLY PAYROLL"
190 PRINT
200 PRINT "NAME";
210 INPUT N$
220 IF N$="END" THEN 800
230 PRINT "EMPLOYEE NUMBER";
240 INPUT N
250 PRINT "GROSS SALARY";
260 INPUT P1
270 PRINT "MARITAL STATUS (M OR S)";
280 INPUT M$
290 PRINT "NUMBER OF EXEMPTIONS";
300 INPUT E
310 GOSUB 400
320 LET P3=P1-(T1+T2+T3)
330 PRINT "FEDERAL TAX=";T3,"STATE TAX=";T1,"LOCAL TAX=";T2
340 PRINT "NET SALARY=";P3
350 GOTO 190
360 DATA 88,133,217,433,583,917,88,183,333,708,1167,1667
370 DATA .14,.17,.20,.18,.21,.24,.14,.17,.16,.19,.21,.25
380 DATA 0,6.3,20.58,63.78,90.78,160.92,0,13.3,38.8,98.8,186.01,291.01
390
400 REM COMPUTATION OF STATE TAX, LOCAL TAX AND ADJUSTED GROSS SALARY
410
420 IF P1>2000 THEN 480
430 IF P1>600 THEN 460
440 LET T1=.01*P1
450 GOTO 490
460 LET T1=.015*(P1-600)
470 GOTO 490
480 LET T1=.02*(P1-2000)
490 IF P1>800 THEN 520
500 LET T2=.01*P1
510 GOTO 530
520 LET T2=8
530 LET P2=P1-54.2*E
540 IF P2>0 THEN 570
550 LET T3=0
560 RETURN
570 GOSUB 600
580 RETURN
590
600 REM      COMPUTATION OF FEDERAL TAX
610
620 LET J=2
630 IF M$="M" THEN 650
640 LET J=1
650 LET T3=0
660 FOR I=1 TO 6
670   IF P2>C(I,J) THEN 710
680   IF I=1 THEN 730
690   LET T3=T(I-1,J)+R(I-1,J)*(P2-C(I-1,J))
700   GOTO 730
710 NEXT I
720 LET T3=T(I,J)+R(I,J)*(P2-C(I,J))
730 RETURN
800 END

```

برنامج بيسك The BASIC Program

يبين شكل ٦ - ١٦ برنامج بيسك كاملاً . لاحظ أول برنامج فرعى ، والذي يحسب الدخل الأساسي المعدل وضريبة الولاية والضريبة المحلية ويتكون من الجمل 400 إلى 580 ، ويتضمن جملي RETURN (الأسطر 560 و 580) . ويستدعى البرنامج الفرعى بواسطة الجملة رقم 310 في الجزء الأساسي من البرنامج .

وتكون الجمل 600 إلى 730 البرنامج الفرعى الثاني ، والذي يستخدم لحساب الضريبة الفيدرالية . ونرى أن هذا البرنامج الفرعى لا يحتوي إلا على جملة RETURN واحدة (السطر 730) ويستدعى هذا البرنامج الفرعى في السطر رقم 570 . (من ثم فإن البرنامج الفرعى متداخلة) . لاحظ أن حلقة FOR-TO التكرارية (السطور من 660 إلى 710) متضمنة في البرنامج الفرعى الثاني .

ومن الممتع أيضاً ملاحظة أن البرنامج يحتوي على جمل READ و INPUT ، وتستخدم READ لإعطاء قيم للمجموعات المتراصة R و T عند بداية تنفيذ البرنامج . بينما جمل INPUT تستخدم لإدخال المعلومات المطلوبة لكل موظف .

يجب أن يفهم القارئ أن هذا البرنامج كان يمكن كتابته بسهولة وبدون استخدام البرامج الفرعية . ومن ثم فالغرض من استخدام البرامج الفرعية في هذا المثال هو تنظيم البرنامج على هيئة عدة أجزاء معرفة تماماً ومستقلة ، ولقد رأينا موقفاً مماثلاً بالنسبة للدوال المعرفة بواسطة المبرمج في مثال ٦ - ١٥ . بينما ، توجد مواقف عديدة يمكن فيها تبسيط البرمجة بطريقة ملحوظة بواسطة استخدام البرامج الفرعية . ويكون هذا صحيحاً خاصة في البرامج التي تحتوي على عدة جمل لاستدعاء نفس البرنامج الفرعى .

وأخيراً ، يبين شكل ٦ - ١٧ مجموعة مخرجات نموذجية لعدد 7 موظفين مختلفين . لاحظ أن كل البيانات (سواء مدخلة أو مخرجة) الخاصة بكل موظف مبنية في كتلة منظمة تماماً ومعنونة بطريقة مرتبة والبيانات المدخلة موضوع تحتها خط .

MONTHLY PAYROLL

```

NAME ?ANDREWS, J. J.
EMPLOYEE NUMBER ?2717
GROSS SALARY ?970.00
MARITAL STATUS (M OR S) ?M
NUMBER OF EXEMPTIONS ?2
FEDERAL TAX=$ 108.984
NET SALARY=$ 742.966
STATE TAX=$ 10.05
LOCAL TAX=$ 8

NAME ?COHEN, A. M.
EMPLOYEE NUMBER ?5375
GROSS SALARY ?1250.00
MARITAL STATUS (M OR S) ?M
NUMBER OF EXEMPTIONS ?3
FEDERAL TAX=$ 170.886
NET SALARY=$ 1055.36
STATE TAX=$ 15.75
LOCAL TAX=$ 9

NAME ?DIPASQUALE, G. V.
EMPLOYEE NUMBER ?4660
GROSS SALARY ?2075.00
MARITAL STATUS (M OR S) ?S
NUMBER OF EXEMPTIONS ?1
FEDERAL TAX=$ 425.832
NET SALARY=$ 1612.67
STATE TAX=$ 28.5
LOCAL TAX=$ 8

NAME ?HOLLAND, C. J.
EMPLOYEE NUMBER ?0872
GROSS SALARY ?520.00
MARITAL STATUS (M OR S) ?S
NUMBER OF EXEMPTIONS ?2
FEDERAL TAX=$ 59.5
NET SALARY=$ 450.1
STATE TAX=$ 5.2
LOCAL TAX=$

NAME ?JONES, D. M.
EMPLOYEE NUMBER ?6839
GROSS SALARY ?1120.00
MARITAL STATUS (M OR S) ?M
NUMBER OF EXEMPTIONS ?2
FEDERAL TAX=$ 156.484
NET SALARY=$ 941.716
STATE TAX=$ 13.8
LOCAL TAX=$ 8

NAME ?KOWALSKI, S.
EMPLOYEE NUMBER ?8462
GROSS SALARY ?1100.00
MARITAL STATUS (M OR S) ?M
NUMBER OF EXEMPTIONS ?3
FEDERAL TAX=$ 142.386
NET SALARY=$ 936.114
STATE TAX=$ 13.5
LOCAL TAX=$ 8

NAME ?LOWE, H. B.
EMPLOYEE NUMBER ?9587
GROSS SALARY ?1075.00
MARITAL STATUS (M OR S) ?M
NUMBER OF EXEMPTIONS ?2
FEDERAL TAX=$ 117.04
NET SALARY=$ 936.835
STATE TAX=$ 13.125
LOCAL TAX=$ 8

NAME ?END

```

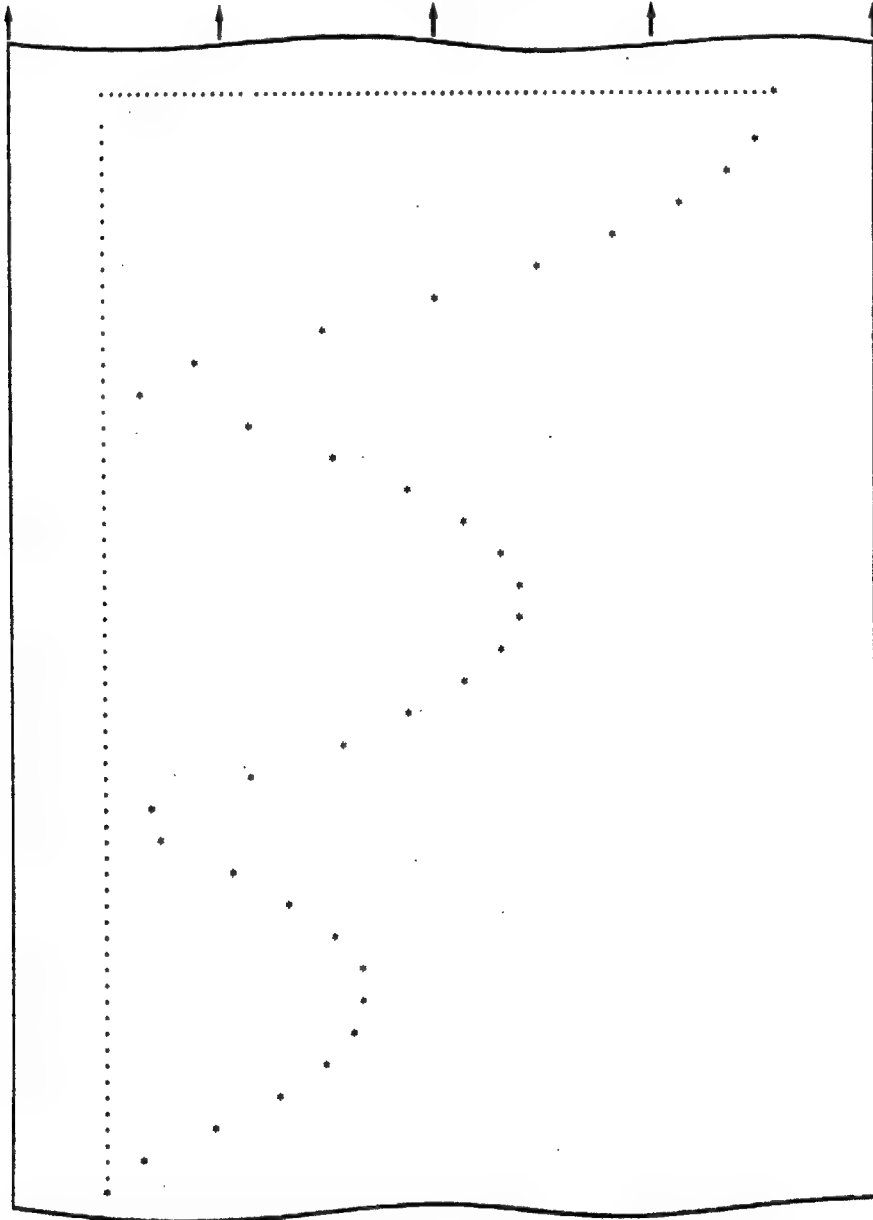
شكل ٦ - ١٧

٦ - ١٠ المخرجات البيانية GRAPHICAL OUTPUT

تولد عدة برامج للحاسب قائمة رقمية من البيانات المخرجة والتي ترسم بيانياً في آخر الأمر على ورق بياني . وفي مثل هذه الحالات يكون غالباً أكثر فاعلية أن يكون الحاسب هو المسئول عن توليد المخرج البياني مباشرة على ورق مثبت على النهاية الطرفية ، وذلك بالإضافة إلى المخرجات الرقمية العادية . ويسمح ذلك للمستخدم بأن يرى المظهر العام للرسم البياني ثم يرجع إلى الجداول الأكثر دقة للبيانات إذا تطلب الأمر ذلك .

إنه لغاية في السهولة إصدار رسم بياني على ورق مثبت على النهاية الطرفية ولتنفيذ ذلك يجب أن نستفيد من استخدام حلقة FOR-TO التكرارية ، والتي تتضمن جملة PRINT محتوية على دالة TAB . الرسم البياني المنتج بهذه الطريقة سوف ينفذ رأسياً متجهاً إلى أسفل الصفحة المطبوعة وفيها المنحنى الحقيقي مثل بعدة نقاط متصلة ومتساوية التباعد وقريبة من بعضها الآخر ، كما هو مبين في شكل ٦ - ١٨ .

الميكانيكية التفصيلية لإصدار هذا الرسم البياني موصفة بواسطة مثال .
اتجاه حركة الورق



شكل ٦ - ١٨

مثال ٦ - ٢٧

لقد تم توليد قائمتين رقميتين Y و T بواسطة برنامج ببسك ، الذى يصف موضع قذيفة في أوقات متعددة . عناصر Y تمثل ارتفاع القذيفة وعناصر T تمثل الأوقات المناظرة (ونرغب في إصدار صورة بيانية لقيم Y مقابل قيم T ، وذلك على ورق مثبت على النهاية الطرفية بمرص 72 حرفاً . ومن أجل توليد رسم بياني واضح بقدر الإمكان سوف نضاعف المسافات بين النقط على المنحنى ، أى ، سوف نترك سطرًا خالياً بين كل نجمتين (*) كما هو مبين في شكل ٦ - ١٨ .

وبما أن الرسم البياني سوف يكون رأسياً بطول الصفحة ، فإن محور Y سوف يولد بواسطة سطر واحد من النقط ومحور T سوف يكون عمودياً على السطر المطبوع . (أى أسفل الصفحة) . من ثم فسوف نطبع نقطة واحدة من محور T بواسطة كل سطر من المخرجات .

دعنا نطبع محور Y مبدئياً موضع القذيفة عند بداية المحور (عند الزمن صفر) . ولعمل ذلك فإننا نطبع نقطة في كل من ال 71 موضع الأولى ثم بعد ذلك نضع النجمة في الموضع رقم 72 وبذلك يمكن كتابة :

```
580 FOR J=0 TO 70
590   PRINT TAB(J);".";
600 NEXT J
610 PRINT TAB(71);"*"
620 PRINT "."
```

سوف تنتج الجمل من 580 إلى 610 محور Y ، كما هو موصوف أعلاه . وسوف يولد سطر متعاقب يحتوى على نقطة واحدة فقط في أول عمود (لتتخذ جزءاً من محور T) وذلك بواسطة الجملة 620 . وهذا السطر مطلوب من أجل الحصول على التباعد المزدوج المطلوب .

بعد ذلك نرغب في طباعة موضع القذيفة في أوقات مختلفة . دعنا نولد سطرين للمخرجات في كل مرة . أولهما يحتوى على نقطة (تمثل محور T) في أول عمود ونجمة (تمثل موضع القذيفة) في وضع ملائم على طول السطر . ويترك السطر التالى خالياً ويحتوى فقط على نقطة في أول موضع . ولإنجاز ذلك يمكننا كتابة :

```
630 FOR I=2 TO I1
640   LET J=INT(71*Y(I)/Y(1))
650   IF J=0 THEN 680
660   PRINT " ";TAB(J);"*"
670   GO TO 690
680   PRINT "*"
690   PRINT " ."
700 NEXT I
```

حيث II رقم صحيح يبين آخر نقطة مطلوب رسمها (آخر وقت) . لاحظ أن أول سطر في الطباعة سوف ينفذ بالجملة من 640 إلى 680 ، والسطر الثانى يولد بواسطة الجملة 690 .

الجملة :

```
640-   LET J=INT(71*Y(I)/Y(1))
```

يمكن أن تتطلب توضيحاً إضافياً أولاً ، لاحظ أن $Y(1)$ هي الارتفاع الحقيق للقذيفة . إذا فرضنا أن ذلك يمثل الارتفاع للقذيفة ، فإن $Y(I)$ وهو ارتفاع القذيفة عند الوقت رقم I سوف يكون رقماً ما محصوراً بين صفر و $Y(1)$ ونبحث عن قيمة للمتغير J رقم صحيح ينحصر ما بين 0 و 71، وسوف تشير قيمة J إلى مكان طباعة النجمة على الورقة ، أى $J=0$ تناظر $Y(I)=0$ و $J=71$ تناظر $Y(I)=Y(1)$ من ثم ينسب مباشرة :

$$\frac{J}{71} = \frac{Y(I)}{Y(1)} \quad \text{or} \quad J = 71 * Y(I) / Y(1)$$

حيث يمكن أن تأخذ J أرقاماً صحيحة فقط ، من ثم نكتب

$$J = \text{INT}(71 * Y(I) / Y(1))$$

في المثال ٦ - ٢٨ سوف نرى برنامج ببسك كاملاً يولد كلا من النتائج الرقية والرسم البياني .

مثال ٦ - ٢٨ محاكاة ارتداد كرة Simulation of a Bouncing Ball

في هذا المثال ، نود حساب حركة كرة من المطاط عند ارتدادها من أعلى إلى أسفل تحت تأثير الجاذبية الأرضية ، بينما في نفس الوقت تسير في الاتجاه الأفقي بسرعة منتظمة . سوف نفرض أن الزحزحة الرأسية المبدئية (H) (أى ، الارتفاع الأساسى أعلى الأرض) والسرعة الأفقية (V) وعدد مرات ارتداد الكرة (N) . وأيضاً سوف يكون معامل الارتداد (C) معلوماً وهو النسبة بين السرعة الرأسية بعد الصدمة مباشرة إلى السرعة الرأسية قبل الصدمة مباشرة .

طريقة إجراء الحسابات Computational Procedure

من أجل حساب موضع الكرة عند أوقات متعددة ، سوف نختار زيادة صغيرة في الوقت (D) ، ثم بعد ذلك نستخدم قوانين الطبيعة التالية التى تطبق مع كل زيادة في الوقت :

$$\begin{aligned} T(I+1) &= T(I) + D \\ X(I+1) &= X(I) + V * D \\ Z(I+1) &= Z(I) - G * D \\ Y(I+1) &= Y(I) + 5 * (Z(I) + Z(I+1)) * D \end{aligned}$$

حيث تشير X إلى الزحزحة الأفقية (تبدأ بصفر عند بداية البرنامج) ، وتشير Z إلى السرعة الرأسية (وهي أيضاً صفر عند بداية البرنامج) و Y هي الارتفاع فوق سطح الأرض و G هي المعجلة الناجمة عن الجاذبية الأرضية (32.2 قدم / ثانية^٢) . تشير الأدلة I و I + 1 لقيم المتغيرات المختلفة عند بداية ونهاية الزيادة في الزمن على الترتيب .

إذا حدث أى ارتداد أثناء الزيادة في الزمن . فن الضرورى تعديل الصيغ الحسابية فشرط الارتداد يميز بقيمة سالبة للمتغير Y(I + 1) والذي يستحيل في التطبيقات الطبيعية . عند حدوث هذا الشرط فإننا نعيد حساب Z(I + 1) و Y(I + 1) كما يلي . أولاً احسب الزمن اللازم لاصطدام الكرة بالأرض ، مبتدئاً من مكانها عند بداية زيادة الزمن للمرة I إذا أطلقنا على هذا الزمن D1 فإنه من التناسب البسيط :

$$\frac{D1}{D} = \frac{Y(I) - 0}{Y(I) - Y(I+1)}$$

حيث يمكن كتابته في البيسك كما يلي

$$D1 = D * Y(I) / (Y(I) - Y(I+1))$$

يمكننا حساب السرعة الرأسية فوراً قبل الاصطدام .

$$Z = Z(I) - G * D1$$

وبذلك فسوف تكون السرعة الرأسية بعد نهاية زيادة الزمن .

$$Z1 = -C * (Z(I) - G * D1)$$

وإلا سوف تكون السرعة الرأسية عند نهاية زيادة الزمن .

$$Z(I+1)=Z1-G*(D-D1)$$

والزحزحة الرأسية عند نهاية زيادة الوقت يمكن كتابتها كما يلي :

$$Y(I+1)=.5*(Z1+Z(I+1))*(D-D1)$$

المخطط التمهيدى للبرنامج The Program Outline

والآن لدينا معلومات كافية لكتابة مخطط تمهيدى لبرنامج يبسك كامل بالتفصيل ،

١ - إقرأ H و V و N و C و D

(أ) إذا كانت $H = 0$ إذن أنه البرنامج

(ب) إذا كانت H لها أى قيمة موجبة فواصل للنقط التالية .

٢ - إبدأ بإعطاء قيم أولية لكل المعاملات :

$$X(1) = 0 \quad I = 1 \text{ (I هو عداد الزيادة)}$$

$$Z(1) = 0 \quad B = 0 \text{ (B هو عداد الارتداد)}$$

$$Y(1) = H \quad T(1) = 0$$

٣ - احسب الزحزحة الرأسية والأفقية والسرعة الرأسية لكل زيادة فى الزمن باستخدام الصيغ الرياضية التى سبق ذكرها .

٤ - إذا اصطدمت الكرة بالأرض أثناء الزيادة فى الزمن ، اختبر لترى هل هذا شرط ارتداد أو إنهاء برنامج .

(أ) شرط ارتداد ($B < N$) - وبذلك أعد حساب السرعة الرأسية والزحزحة الرأسية لتأخذ فى الاعتبار الارتداد ، ثم زد عداد الارتداد (أى $B = B + 1$) ثم بعد ذلك أكل إلى الزيادة التالية فى الزمن .

(ب) شرط الانتهاء ($B = N$) - أوجد الزمن النهائى والزحزحة الأفقية عندما تصطدم الكرة بالأرض .

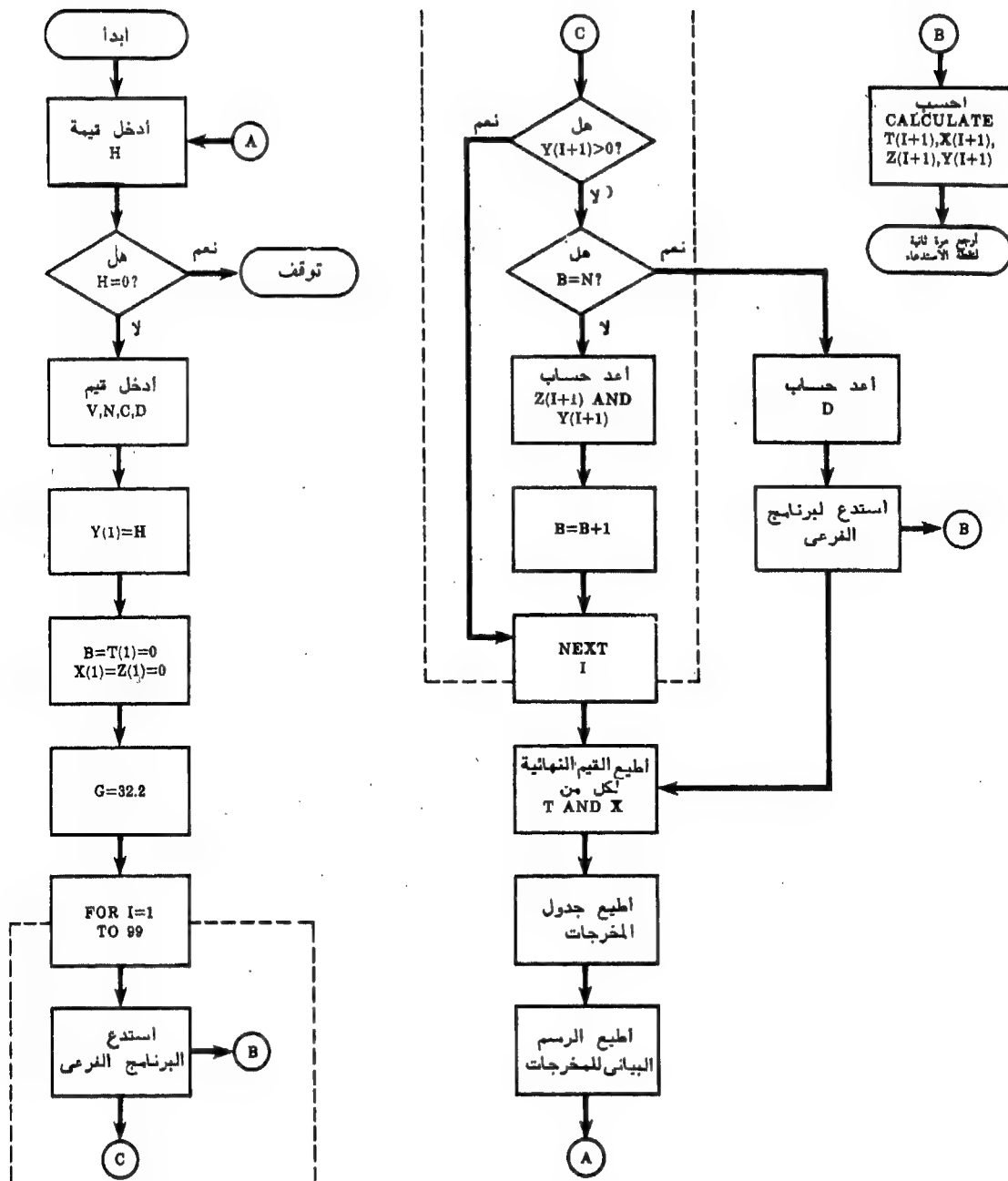
٥ - اطبع القيم النهائية لكل من X و T يليها جدول كامل لكل من T و X و Y و Z .

٦ - ارسم Y مقابل T باستخدام الطريقة التى تمت مناقشتها فى المثال ٦ - ٢٧ .

٧ - ارجع مرة ثانية للخطوة ١ .

يبين شكل ٦ - ١٩ . خريطة سير عمليات كاملة لهذه الحسابات . لاحظ أن الحسابات لكل من $T(I + 1)$ و $X(I + 1)$ و $Z(I + 1)$ و $Y(I + 1)$ مبنية بداخل هيكل البرنامج الفرعى .

يجب أن يكتشف القارئ الذى درس التكامل بالطرق العددية أننا فى هذا المثال نكامل المعادلة التفاضلية من الدرجة الثانية $d^2y/dt^2 = -g$ وتعرف طريقة التكامل باسم طريقة أويلر المعدلة *modified Euler method*



```

10 REM          SIMULATION OF A BOUNCING BALL
20 DIM X(100),Y(100),Z(100),T(100)
30 PRINT "INITIAL HEIGHT OF BALL (FT)";
40 INPUT H
50 IF H=0 THEN B10
60 PRINT "INITIAL HORIZONTAL VELOCITY (FT/SEC)";
70 INPUT V
80 PRINT "NUMBER OF BOUNCES";
90 INPUT N
100 PRINT "BOUNCE COEFFICIENT";
110 INPUT C
120 PRINT "LENGTH OF TIME INCREMENT (SEC)";
130 INPUT D
140 PRINT
150
160 REM          INITIALIZE PARAMETERS
170
180 LET B=T(1)=X(1)=Z(1)=0
190 LET Y(1)=H
200 LET G=32.2
210
220 REM COMPUTE VELOCITY AND DISPLACEMENT FOR EACH TIME INCREMENT
230
240 FOR I=1 TO 99
250   GOSUB 730
260   IF Y(I+1)>0 THEN 330
270   IF B=N THEN 360
280   LET D1=D*Y(I)/(Y(I)-Y(I+1)) 'CORRECT FOR BOUNCE CONDITION
290   LET Z1=-C*(Z(I)-G*D1)
300   LET Z(I+1)=Z1-G*(D-D1)
310   LET Y(I+1)=.5*(Z1+Z(I+1))*(D-D1)
320   LET B=B+1
330 NEXT I
340 GOTO 400
350
360 REM          BALL HITS GROUND FOR LAST TIME
370
380 LET D=D*Y(I)/(Y(I)-Y(I+1))
390 GOSUB 730
400 LET I1=I+1
410 LET T1=T(I1)
420 LET X1=X(I1)
430
440 REM          PRINT NUMERICAL OUTPUT
450
460 PRINT "HORIZONTAL DISTANCE TRAVELED=";X1;"FT"
470 PRINT "TIME REQUIRED=";T1;"SECS"
480 PRINT
490 FOR I=1 TO I1
500   PRINT "T=";T(I),"X=";X(I),"Y=";Y(I),"Z=";Z(I)
510 NEXT I
520 PRINT
530
540 REM          PRINT GRAPHICAL OUTPUT
550
560 PRINT "GRAPHICAL SOLUTION TO BOUNCING BALL PROBLEM"
570 PRINT
580 FOR J=0 TO 70
590   PRINT TAB(J);". ";
600 NEXT J
610 PRINT TAB(71);"*"
620 PRINT ". "
630 FOR I=2 TO I1 'GENERATE SUCCESSIVE POINTS OF CURVE
640   LET J=INT(71*Y(I)/Y(1))
650   IF J=0 THEN 680
660   PRINT ". ";TAB(J);"*"
670   GOTO 690
680   PRINT "*"
690   PRINT ". "
700 NEXT I
710 GOTO 30
720
730 REM SUBROUTINE TO CALCULATE VELOCITY AND DISPLACEMENT AT END
740 REM          OF TIME INCREMENT
750
760 LET T(I+1)=T(I)+D
770 LET X(I+1)=X(I)+V*D
780 LET Z(I+1)=Z(I)-G*D
790 LET Y(I+1)=Y(I)+.5*(Z(I+1)+Z(I))*D
800 RETURN
810 END

```

برنامج بيسك The BASIC Program

يبين شكل ٦ - ٢٠ برنامج بيسك كاملاً للقيام بهذه العمليات الحسابية . يسمح البرنامج بعدد من الزيادات المتتالية للزمن تصل إلى 100 مرة ، وبذلك فإن طول فترة الزيادة الزمنية (D) يجب أن تختار كبيرة نسبياً حتى لا تتعدى عدد الزيادات الزمنية هذا الرقم . (ومن الناحية الأخرى . فإن D لا يمكن أن تعطى قيمة كبيرة وألا أصبح من غير الممكن تطبيق الصيغة الرياضية التي تحسب $T(I+1)$ و $X(I+1)$ و $Y(I+1)$ و $Z(I+1)$. وكقاعدة تقريبية يمكن أن يمثل كل ارتداد بواسطة 8 إلى 20 نقطة) .

والبرنامج الفرعى الذى يستخدم فى حساب $T(I+1)$ و $X(I+1)$ و $Y(I+1)$ و $Z(I+1)$ يتكون من الجمل 730 إلى 800 ويجب أن يكون مفهوماً أن هيكل البرنامج الفرعى ليس جوهرياً للقيام بهذه الحسابات . وأن استخدام البرنامج الفرعى يكون مرغوباً فيه حيث أن هذه الكتلة من الجمل قد استندت من مكانين مختلفين بداخل البرنامج (جملتى 250 و 390) .

توليد جدول البيانات المخرجة ينفذ بطريقة مباشرة بواسطة الحلقة التكرارية FOR - TO (الجمل 490 إلى 510) . بالرغم من أن الطريقة المستخدمة لتوليد الرسم البياني (الجمل 560 إلى 700) أقل وضوحاً ، فهذا الجزء من البرنامج مطابق تماماً لما سبقته مناقشته فى المثال ٦ - ٢٧ . ومن ثم فإن المنطق المتبع يجب أن يكون واضحاً .

الشكل ٦ - ٢١ (أ) : يحتوى على المخرجات العددية التي تم توليدها لبيانات الإدخال التالية : (والبيانات المدخلة موضوع تحتها خط) .

$$H = 2.00 \text{ ft}, V = 1.20 \text{ ft/sec}, N = 3, C = 0.80, D = 0.05 \text{ sec}$$

يتضح أن المسافة المطلوبة حتى تكمل الكرة ثلاثة ارتدادات كاملة هي 2.06 قدم والزمن المناظر لذلك هو 1.72 ثانية .

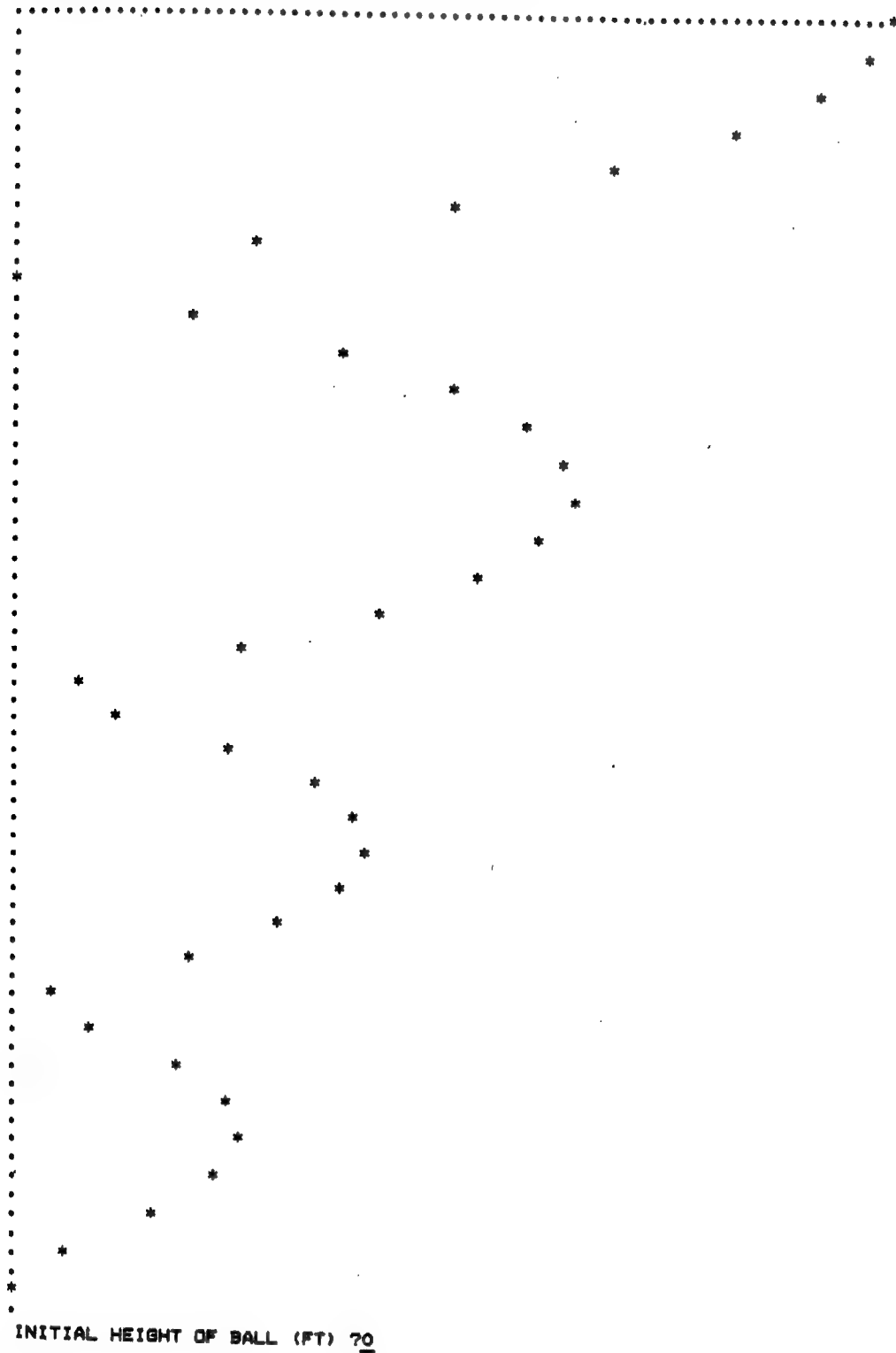
INITIAL HEIGHT OF BALL (FT) 72.00
INITIAL HORIZONTAL VELOCITY (FT/SEC) 71.20
NUMBER OF BOUNCES 73
BOUNCE COEFFICIENT 70.80
LENGTH OF TIME INCREMENT (SEC) 70.05

HORIZONTAL DISTANCE TRAVELED= 2.06344 FT
TIME REQUIRED= 1.71953 SECS

T= 0	X= 0	Y= 2	Z= 0
T= 0.05	X= 6.00000E-2	Y= 1.93973	Z=-1.61
T= 0.1	X= 0.12	Y= 1.839	Z=-3.22
T= 0.15	X= 0.18	Y= 1.63773	Z=-4.83
T= 0.2	X= 0.24	Y= 1.336	Z=-6.44
T= 0.25	X= 0.3	Y= 0.99373	Z=-8.05
T= 0.3	X= 0.36	Y= 0.551	Z=-9.66
T= 0.35	X= 0.42	Y= 2.77500E-2	Z=-11.27
T= 0.4	X= 0.48	Y= 0.396269	Z= 7.3392
T= 0.45	X= 0.54	Y= 0.732979	Z= 5.9292
T= 0.5	X= 0.6	Y= 0.989189	Z= 4.3192
T= 0.55	X= 0.66	Y= 1.1649	Z= 2.7092
T= 0.6	X= 0.72	Y= 1.26011	Z= 1.0992
T= 0.65	X= 0.78	Y= 1.27482	Z=-0.5108
T= 0.7	X= 0.84	Y= 1.20903	Z=-2.1208
T= 0.75	X= 0.9	Y= 1.06274	Z=-3.7308
T= 0.8	X= 0.96	Y= 0.835949	Z=-5.3408
T= 0.85	X= 1.02	Y= 0.528659	Z=-6.9508
T= 0.9	X= 1.08	Y= 0.140869	Z=-8.5608
T= 0.95	X= 1.14	Y= 0.233292	Z= 6.1104
T= 1.	X= 1.2	Y= 0.498562	Z= 4.5004
T= 1.05	X= 1.26	Y= 0.683332	Z= 2.8904
T= 1.1	X= 1.32	Y= 0.787602	Z= 1.2804
T= 1.15	X= 1.38	Y= 0.811372	Z=-0.329597
T= 1.2	X= 1.44	Y= 0.754642	Z=-1.9396
T= 1.25	X= 1.5	Y= 0.617413	Z=-3.5496
T= 1.3	X= 1.56	Y= 0.399683	Z=-5.1596
T= 1.35	X= 1.62	Y= 0.101453	Z=-6.7696
T= 1.4	X= 1.68	Y= 0.189303	Z= 4.58198
T= 1.45	X= 1.74	Y= 0.378152	Z= 2.97198
T= 1.5	X= 1.8	Y= 0.486501	Z= 1.36198
T= 1.55	X= 1.86	Y= 0.514351	Z=-0.248016
T= 1.6	X= 1.92	Y= 0.4617	Z=-1.85802
T= 1.65	X= 1.98	Y= 0.328549	Z=-3.46802
T= 1.7	X= 2.04	Y= 0.114898	Z=-5.07802
T= 1.71953	X= 2.06344	Y= 9.58084E-3	Z=-5.7069

شكل ٦ - ٢١ (أ)

GRAPHICAL SOLUTION TO BOUNCING BALL PROBLEM



شکل ۶-۲۱ (ب)

نرى في شكل ٦ - ٢١ (ب) رسماً بيانياً مجهزاً بالحاسب موضحاً أماكن الكرة في أزمنة متعددة . ويمكن أن ترى الارتدادات الفردية بكل وضوح في هذا الشكل .

وأخيراً نرى في نهاية شكل ٦ - ٢١ (ب) طلباً لمجموعة جديدة من بيانات الإدخال . وتتوقف الحسابات إذا أعطينا قيمة صفر للمتغير H وسوف نتعرض لتوليد المخرجات البيانية بشكل أوسع في الفصل الثاني عشر الذي يختص ببيانات الحاسب الدقيق .

اسئلة للمراجعة Review Questions

- ٦ - ١ ماهي أوجه الاختلاف بين الدالة والبرنامج الفرعي ؟
- ٦ - ٢ هل الدوال والبرامج الفرعية دائماً مطلوبة في برامج بيسك ، ماهي المزايا في استخدامها ؟
- ٦ - ٣ ماهي الطريقة التي يمكن أن تسهم بها الدوال والبرامج الفرعية في تطوير تنظيم برنامج بيسك .
- ٦ - ٤ لخص قواعد تسمية الدوال . وكيف يمكن تمييز الدوال الرقية عن الدوال الحرفية ؟
- ٦ - ٥ ماهو الفرض من جملة DEF ؟ وكيف تكتب ؟
- ٦ - ٦ ماهي القواعد التي تحكم استخدام الخلاصات في النوال ؟ وهل يمكن للدالة أن تستخدم متغيرات غير موصوفة كخلاصات بها ؟
- ٦ - ٧ ماهو الفرق بين تعريف الدالة والإشارة إليها ؟ وكيف يمكن الإشارة لدالة ؟
- ٦ - ٨ ماهي القيم الزائفة ؟ وما هو التناظر الذي يجب وجوده بين مجموعة من الخلاصات عند الإشارة للدالة والخلاصات الزائفة المتزاملة لها ؟
- ٦ - ٩ هل يمكن لخلاصة أن تتكون من أي شيء خلاف متغير ليس له دليل (على سبيل المثال : الثابت ، المتغير ذو الدليل أو الصيغة الرياضية) ؟ هل هذا صحيح أيضاً لخلاصة زائفة ؟
- ٦ - ١٠ كيف تكتب جملة DEF في دالة متعددة الأسطر ؟
- ٦ - ١١ اذكر مكانين يجب أن يظهر فيهما اسم الدالة المتعددة الأسطر ؟
- ٦ - ١٢ ماهو الفرض من جملة FNEND ؟ وكيف تكتب ؟
- ٦ - ١٣ هل يمكن تحويل التحكم خارج الدالة متعددة الأسطر بواسطة جملة GO TO أو جملة RETURN ؟
- ٦ - ١٤ هل يجب أن تكون القيمة الناتجة من الدالة من نفس نوع خلاصات الدالة ؟
- ٦ - ١٥ هل يجب أن تكون كل خلاصات الدالة من نفس النوع ؟
- ٦ - ١٦ كيف تخزن الحروف بداخل الحاسب ؟
- ٦ - ١٧ ما هو كود ASCII ذو الأرقام السبعة الثنائية ؟
- ٦ - ١٨ ماهو الفرض من جملة CHANGE ؟ اذكر طريقتين يمكن أن تكتب بهما ؟

- ٦ - ١٩ ماهو الغرض من دالة ASC ؟ وكيف تستخدم ؟
- ٦ - ٢٠ ماهو الغرض من دالة CHR\$ ؟ وكيف تستخدم ؟
- ٦ - ٢١ ماهو الغرض من دالة RND ؟ وكيف تستخدم ؟ وهل تتطلب هذه الدالة خلاصة ؟
- ٦ - ٢٢ ماهو المقصود بالأرقام العشوائية الزائفة ؟ وكيف تختلف الأرقام العشوائية الزائفة عن الأرقام التي هي في حقيقة الأمر عشوائية ؟
- ٦ - ٢٣ ماهو الغرض من جملة RANDOMIZE ؟ وكيف تكتب ؟
- ٦ - ٢٤ لخص القواعد الخاصة بتعريف البرامج الفرعية . هل يجب أن يبدأ البرنامج الفرعي بجملة معينة ؟
- ٦ - ٢٥ هل يمكن تضمين خلاصات في البرنامج الفرعي ؟
- ٦ - ٢٦ هل يمكن أن ينتهي برنامج فرعي بجملة FNEND ؟ أو جملة RETURN ؟ أو جملة END ؟
- ٦ - ٢٧ ماهو الغرض من جملة RETURN ؟ وكيف تكتب ؟ ماذا يحدث عند مقابلة جملة RETURN أثناء تنفيذ البرنامج ؟
- ٦ - ٢٨ هل يمكن أن يحتوى البرنامج الفرعي على أكثر من جملة RETURN ؟ إشرح .
- ٦ - ٢٩ ماهو الغرض من جملة GOSUB ؟ وكيف تكتب ؟ هل يمكن لبرنامج يحتوى على برنامج فرعي واحد أن يحتوى على أكثر من جملة GOSUB ؟
- ٦ - ٣٠ هل يمكن تضمين حلقة تكرارية FOR—TO في برنامج فرعي أو دالة متعددة الأسطر ؟
- ٦ - ٣١ هل يمكن تحويل التحكم إلى خارج البرنامج الفرعي بواسطة جملة GO TO ؟ أو جملة IF—THEN ؟
- ٦ - ٣٢ صف الترتيب الهرمي الذي يجب أن يلاحظ عند تداخل البرامج الفرعية ؟
- ٦ - ٣٣ ماهي مزايا عرض البيانات المخرجة بيانياً ؟
- ٦ - ٣٤ ماهي الدوال المكتوبة التي تستخدم لإنتاج مخرجات بيانية ؟ وبأي نوع من البرمجة الهيكلية يمكن تضمين هذه الدوال عادة ؟

مسائل محلولة

Solved Problems

- ٦ - ٣٥ اكتب دالة ببسك لكل من المواقف التالية :

$$z = \frac{(u/v) + (x/y)}{2} \quad (أ) \text{ احسب قيمة الصيغة الجبرية :}$$

10 DEF FNZ(U,V,X,Y)=(U/V+X/Y)/2

- (ب) إذا كانت X ممثلة بكسرية موجبة ، أوجد قيمة مقربة للقيمة X لها رقمان عشريان يمين العلامة العشرية .

20 DEF FNY(X)=.01*INT(100*(X+.005))

$$p = \begin{cases} \log(t^2 - a) & \text{for } t^2 > a \\ \log(t^2) & \text{for } t^2 \leq a \end{cases} \quad (ج) \text{ احسب قيمة الصيغة الجبرية :}$$

```

30 DEF FNP(T,A)
40 IF T<=A THEN 70
50 LET FNP=LOG(T+2-A)
60 GO TO 80
70 LET FNP=LOG(T+2)
80 FNEND

```

(د) احسب مجموع الـ N عنصراً الأول في قائمة رقمية L ، أى احسب مجموع $L(1) + L(2) + L(3) + \dots + L(N)$

```

100 DEF FNS(N)
110 LET S=0
120 FOR I=1 TO N
130   LET S=S+L(I)
140 NEXT I
150 LET FNS=S
160 FNEND

```

(هـ) نفرض أن كلا من M\$ و N\$ تمثل حرفاً واحداً . أنشئ سلسلة حرفية واحدة تحتوى هذين الحرفين مرتبة ترتيباً أبجدياً .

```

200 DEF FNN$(M$,N$)
210 LET L(0)=2
220 IF M$>N$ THEN 260
230 LET L(1)=ASC(M$)
240 LET L(2)=ASC(N$)
250 GO TO 280
260 LET L(1)=ASC(N$)
270 LET L(2)=ASC(M$)
280 CHANGE L TO L$
290 LET FNN$=L$
300 FNEND

```

(و) احسب متوسط رقين عشوائيين كل منهما له قيمة تتراوح ما بين A و B .

```

100 DEF FNR(A,B)
110 LET R1=A+(B-A)*RND
120 LET R2=A+(B-A)*RND
130 LET FNR=(R1+R2)/2
140 FNEND

```

يمكن أن تكتب هذه الدالة أيضاً

```

100 DEF FNR=A+(B-A)*(RND+RND)/2

```

٦ - ٣٦ كل من المواقف الموصوفة التالية تتطلب استدعاء إحدى الدوال المعرفة في المسألة ٦ - ٣٥ . اكتب جملة ييسر لك المناسبة أو مجموعة من الجمل المتتالية ، في كل حالة .

(أ) اطبع قيمة f ، حيث
$$f = \frac{(a/b) + (c/d)}{2}$$

(أنظر المسألة ٦ - ٣٥ (أ))

100 PRINT FNZ(A,B,C,D)

(ب) نفرض أن T تمثل كمية موجبة يمكن أن تزيد قيمتها عن الواحد احسب قيمة $T1$ ، حيث $T1$ لها نفس قيمة T ماعداً أن الجزء الكسرى في $T1$ يقرب إلى أقرب رقمين عشريين .

(أنظر المسألة ٦ - ٣٥ (ب)) .

110 LET T1=INT(T)+FNY(T-INT(T))

(-) دع $P1$ تمثل الكمية

$\log[(a+b)^2 - c]$ if $(a+b)^2 > c$

$\log[(a+b)^2]$ if $(a+b)^2 \leq c$

(أنظر المسألة ٦ - ٣٥ (-)) .

30 LET P1=FNP((A+B),C)

(د) قائمة عددية L بها 101 عنصراً . فبتدنا بالعنصر (١) L ، حدد كم عدد العناصر المتعاقبة التي يمكن إضافتها بدون أن يتعدى المجموع القيمة 25 (انظر مسألة ٦ - ٣٥ (د)) .

40 FOR J=1 TO 100
50 IF FNS(J)>25 THEN 80
60 NEXT J
70 LET J=101
80 PRINT "N=";J-1

(هـ) كل من المتغيرات M و N تمثل مكاناً كود ASCII . لحرف واحد . كون سلسلة حرفية تحتوي على حرفين بترتيب أبجدي (انظر المسألة ٦ - ٣٥ (هـ)) .

80 LET L\$=FNN\$(CHR\$(M),CHR\$(N))

(و) حدد متوسط رقمين عشوليين . كل منهما له قيمة ما بين 1 و 10 . حول التحكم إلى الجملة رقم 250 إذا كان المتوسط يتعدى 5 . (انظر المسألة ٦ - ٣٥ (و)) .

100 IF FNR(1,10)>5 THEN 250

٦ - ٣٧ تحتوي كل من المسائل التالية على تعريف دالة و / أو استدعاء دالة حيث كتبت بصورة غير صحيحة . تعرف على كل الأخطاء .

10 DEF FNW(A,B,C↑2,3)=((A+B)*C↑2)/3 (أ)

لا يمكن استخدام الثوابت والصيغ الرياضية كخلاصات زائفة .

```
10 DEF FNC(T1,T2,N)=((T1-T2)/T2)↑N
```

(ب)

```
60 LET V=C*FNC(2*A,F$)
```

لا تتوافق الخلاصات في الإشارة إلى الدالة مع الخلاصات الزائفة في تعريف الدالة من حيث العدد والنوع .

```
10 DEF FN4(X(1),X(2),X(3))=X(1)+2*X(2)-3*X(3)
```

تحتوي هذه الجملة على خطأين :

(i) اسم الدالة غير صحيح .

(ii) لا يمكن أن تظهر المتغيرات ذات الأدلة كخلاصات زائفة .

```
10 DEF FNG(A,B,C)
```

```
20 LET P=A+B*X+C*X↑2
```

```
30 LET Q=B+C*X
```

```
40 LET G=P+Q*X+C*X↑2
```

```
50 FNEND
```

اسم الدالة FNG لم تحدد أى قيمة بداخل الدالة .

```
100 DEF FNC(X,Y,Z)
```

```
110 IF X+Y>Z THEN 140
```

```
120 LET FNC=LOG(Z-(X+Y))
```

```
130 RETURN
```

```
140 LET FNC=LOG(Z)
```

```
150 FNEND
```

لا يمكن تحويل التحكم خارج الدالة بواسطة جملة RETURN

٦ - ٣٨ مجموعة الجمل التالية تمثل اجزاء من برامج بيسك تحتوي على برنامج فرعى أو أكثر . وقد كتب مثال منها بصورة صحيحة .

```
10 DIM L(100)
```

(أ)

```
...
```

```
60 GOSUB 200
```

```
...
```

```
120 GOSUB 200
```

```
...
```

```
160 GOSUB 200
```

```
...
```

```
190 STOP
```

```
200 LET S=0
```

```
210 FOR I=1 TO N
```

```
220 LET S=S+L(I)
```

```
230 NEXT I
```

```
240 RETURN
```

```
250 END
```

برنامج فرعى

لاحظ أن البرنامج الفرعى قد تم استدعاؤه من ثلاث نقاط مختلفة بداخل برنامج الحاسب .

```

50 GOSUB 120
...
120 IF A>B THEN 150
130 LET C=SQR((B-A)↑N)
140 RETURN
150 LET C=SQR(((A+B)/(A-B))↑N)
160 RETURN

```

(ب)

برنامج فرعى

يحتوى البرنامج الفرعى فى هذا المثال على جملة RETURN

```

10 DEF FNZ(X,Y)=X↑2+Y↑2
...
70 GOSUB 180
...
180 REM SAMPLE SUBROUTINE
...
210 LET W=FNZ(A,B+C)
...
250 RETURN

```

(ج)

برنامج فرعى

لاحظ أن البرنامج الفرعى يستدعى الدالة FNZ المعرفة بواسطة المبرمج .

```

75 GOSUB 300
...
125 GOSUB 200
...
200 LET Z=C1*X+C2*Y
...
250 GOSUB 300
...
290 RETURN
300 LET W=(U+V)/Z
...
370 RETURN
380 END

```

(د)

البرنامج الفرعى الأول

البرنامج الفرعى الثانى

يستخدم هذا المثال البرامج المساعدة المتداخلة. لاحظ أن البرنامج الفرعى الثانى يستدعى من البرنامج الفرعى الأول ومن الجزء الأساسى من البرنامج .

٦ - ٣٩ تمثل المجموعات التالية من الجمل أجزاء من برامج يسلك والى تحتوى على برامج فرعية . كل مثال يحتوى على خطأ أو أكثر . تعرف على كل الأخطاء .

```

45 GOSUB 165
...
165 LET C=C1+C2+C3
...
190 GO TO 60
...
225 RETURN
230 FNEND
235 END

```

(أ)

البرنامج الفرعى

يحتوى هذا البرنامج على خطأين :

(i) لا يمكن تحويل التحكم خارج البرنامج الفرعى بواسطة جملة . GO TO

(ii) لا يمكن أن ننهى البرنامج الفرعى بجملة . FNEND

60 GOSUB 200

(ب)

...

120 IF X<Y THEN 225

...

200 REM START OF SUBROUTINE

...

300 RETURN

} البرنامج الفرعى

. IF-THEN يمكن تحويل التحكم داخل البرنامج الفرعى بواسطة جملة

30 GOSUB 100

(ج)

...

100 REM SUBROUTINE A

...

120 GOSUB 200

...

160 RETURN

200 REM SUBROUTINE B

...

225 GOSUB 100

...

245 RETURN

} البرنامج الفرعى A

} البرنامج الفرعى B

البرامج الفرعية ليست متداخلة تداخلاً صحيحاً (يستدعى البرنامج الفرعى A البرنامج الفرعى B وهو بالتالى

يستدعى البرنامج الفرعى A) .

٦ - ٤٠ اكتب جزءاً من برنامج يولد رسماً بيانياً للدالة $y = \sin t$ ويولد 130 نقطة متباعدة بمسافات متضاعفة ، حيث المحور t يتجه إلى وسط الصفحة المطبوعة . دع الزيادة في الزمن تكون بمقدار 0.1 ثانية .

```

10 FOR J=0 TO 70
20   IF J=35 THEN 50
30   PRINT TAB(J);".";
40   GO TO 60
50   PRINT TAB(J);"*";
60 NEXT J
70 PRINT
80 PRINT TAB(35);"."
90 LET T=0
100 FOR I=2 TO 130
110   LET T=T+.1
120   LET J=35+INT(35*SIN(T))
130   IF J>35 THEN 190
140   IF J=35 THEN 170
150   PRINT TAB(J);"*";TAB(35);"."
160   GO TO 200
170   PRINT TAB(J);"*"
180   GO TO 200
190   PRINT TAB(35);".";TAB(J);"*"
200   PRINT TAB(35);"."
210 NEXT I
220 END

```

مسائل تكميلية

Supplementary Problems

٦ - ٤١ اكتب دالة ببسك لحساب قيمة كل من الصيغ الجبرية المبينة فيما يلي :

$$y = ax^b \quad (أ)$$

$$q = c_0 + c_1r + c_2r^2 + c_3r^3 + c_4r^4 \quad (ب)$$

$$i = (j+k)^{j+k} \quad (ج)$$

$$r = \begin{cases} \sqrt{b^2 - 4ac} & \text{if } b^2 > 4ac \\ \sqrt{4ac - b^2} & \text{if } b^2 < 4ac \end{cases} \quad (د)$$

٦ - ٤٢ اكتب دالة ببسك لكل من المواقف الموصوفة فيما يلي :

(أ) إذا كانت Z تمثل كمية موجبة يمكن أن تمتد قيمتها الواحد الصحيح ، أوجد قيمة صحيحة مقربة .

(ب) احسب حاصل ضرب عدد N من العناصر الأولى من قائمة رقمية T ، أى احسب حاصل الضرب $T(1) * T(2) * \dots * T(N)$

(ج) ولد 5 أرقاماً عشوائية كل منها قيمته بين A و B ، حيث A و B تمثل كميات موجبة و $A < B$ والنتيجة هي أكبر قيمة .

(د) افحص إشارة الرقم الممثل بالمتغير X ، إذا كانت X سالبة فالنتيجة هي NEGATIVE وإذا كانت X موجبة فالنتيجة هي POSITIVE أما إذا كانت قيمة X هي صفر فالنتيجة هي ZERO .

(هـ) نفرض أن N\$ تمثل كلمة متعددة الحروف . افحص كل حرف من الحروف والنتيجة هي الحرف الذى يظهر أول الحروف الهجائية .

٦ - ٤٣ كل من المواقف الموصوفة فيما يلي تتطلب استدعاء دالة معرفة في المسألة ٦ - ٤١ أو ٦ - ٤٢ . اكتب جملة ببسك ملائمة أو مجموعة متتالية من الجمل في كل حالة .

$$(أ) \text{ احسب قيمة } t = (c_1 + c_2)(x+y)^3 \quad (\text{أنظر مسألة ٦ - ٤١ (أ)}) .$$

$$(ب) \text{ احسب قيمة } q = c_0 + c_1 \log(x) + c_2 \log(x)^2 + c_3 \log(x)^3 + c_4 \log(x)^4 \quad (\text{أنظر مسألة ٦ - ٤١ (ب)}) .$$

$$(ج) \text{ اطيح قيمة } f = (a-b+c)^{a-b+c} \quad (\text{أنظر مسألة ٦ - ٤١ (ج)}) .$$

(د) احسب الفرق بين رقم ممثل بالمتغير X وأقرب رقم صحيح له . عبر عن هذا الفرق بكمية موجبة . (أنظر المسألة ٦ - ٤٢ (أ)) .

(هـ) قائمة رقمية تحتوى على 61 عنصراً . مبتدئاً بالعنصر T(1) ، قرر كم عدد العناصر المتتالية التى يجب أن تضرب فى بعضها من أجل أن يمتد حاصل الضرب الكمية 1000 نفرض أن كل الكميات موجبة . (أنظر المسألة ٦ - ٤٢ (ب)) .

(و) ولد 20 مجموعة كل منها بها 5 أرقام عشوائية ، وكل من هذه الأرقام قيمتها تتراوح ما بين 2 و 5 . اطيح أكبر رقم عشوائي تم الحصول عليه في كل مجموعة من هذه المجموعات المكونة من 5 أرقام . (أنظر المسألة ٦ - ٤٢ (ج)) .

٦ - ٤٤ تبين كل من المسائل التالية جزءاً من برنامج بيסק يحتوى على دالة أو برنامج فرعى . يوجد على الأقل خطأ واحد في كل حالة تعرف على كل الأخطاء .

10 DEF FNK(J,K)=(C1*J+C2*K)/(J+K) (أ)

60 LET T=FNK(A,B,C)

10 DEF FNC(X,Y) (ب)

20 IF X<Y THEN 50

30 LET C=SQR((X-Y)/2)

40 RETURN

50 LET C=SQR(X/(X+Y))

60 RETURN

70 FNEND

50 GOSUB 200 (ج)

...

80 GO TO 230

...

200 REM SUBROUTINE A

...

230 LET Z=X+Y

...

250 RETURN

260 FNEND

} برنامج فرعى

10 DEF FNZ1(A+2,B+2) (د)

...

50 LET FNZ1=(A+2-B+2)/(A+2+B+2)

60 FNEND

10 DEF FNK(J,K)=(C1*J+C2*K)/(J+K) (هـ)

...

80 PRINT J,K,FNK

100 GOSUB 200 (و)

...

200 REM SUBROUTINE A

...

240 IF D<.01 THEN 150

...

270 RETURN

} برنامج فرعى

10 DEF FNX(A,B,C) (ز)

...

50 GOSUB 300

...

80 FNEND

300 REM FIRST SUBROUTINE

...

330 LET Y=FNX(U,V,W)

...

350 RETURN

} برنامج فرعى

مسائل للبرمجة Programming Problems

٦ - ٤٥ عدل البرنامج المبين في مثال ٦ - ٦ لإيجاد أقل قيمة لدالة معينة. استخدم البرنامج لتحصل على جذور المعادلات التالية ، باستخدام الطريقة الموصوفة في نهاية مثال ٦ - ٦ .

$$x + \cos x = 1 + \sin x, \pi/2 < x < \pi \quad (أ)$$

$$x^5 + 3x^2 = 10, 0 \leq x \leq 3 \quad (ب) \text{ (أنظر مثال ٤ - ٥)}$$

٦ - ٤٦ عدل البرنامج المبين في مثال ٦ - ١٥ مع استبدال الدالة FNP ببرنامج فرعى .

٦ - ٤٧ عدل البرنامج المبين في مثال ٦ - ٢٠ حتى يمكن محاكاة ألعاب الكراسي المتتالية أوتوماتيكياً ، بطريقة غير تخطئية ضمن عداد يقرر مجموع مرات الفوز ، ومتغير للمدخلات سوف تتعدد قيمته عدد مرات اللعب التي سنقوم بمحاكاتها .

استخدم البرنامج لمحاكاة عدد كبير من الألعاب (مثل ، 1000) . قدر احتمالات الفوز عند لعب الكراسي (تمثل هذه القيمة برقم عشرى مساو لعدد مرات الفوز مقسوماً على إجمالى عدد الألعاب . إذا تعدت الاحتمالات 0.500 ، فهى في صالح اللاعب ، وإلا فهى في غير صالحه) .

٦ - ٤٨ عدل البرنامج المبين في مثال ٦ - ٢٦ لتشغيل المرتبات الأسبوعية . استعمل دالة بدلا من برنامج فرعى وذلك لحساب مقدار الضريبة الفيدرالية المستقطعة .

مقدار ضريبة الدخل الفيدرالية المستقطعة على أساس أسبوعى مبينة في جدول ٦ - ٣ . هذه الأرقام مبينة على أساس إجمالى الدخل الأسبوعى المعدل ، وهو مساو لإجمالى الدخل الأسبوعى منقوصاً منه \$35.58 لكل معول .

جدول ٦ - ٣ نسبة استقطاعات ضريبة الدخل الفيدرالية

فترة المرتبات الأسبوعية					
نسخة أعزب متضمناً ذلك رب الأسرة			الأشخاص المتزوجون		
مقدار ضريبة الدخل التي سوف تستقطع هي :			مقدار ضريبة الدخل التي سوف تستقطع هي :		
إذا كان إجمالى الأجر هو			إذا كان إجمالى الأجر هو :		
لا تزيد عن \$20			لا تزيد عن \$20		
0			0		
ولكن ليست أكثر من			ولكن ليست أكثر من		
النسبة التي تزيد عن			النسبة التي تزيد عن		
\$20	-\$31	14%	\$20	-\$42	14%
\$31	-\$50	\$1.54 plus 17%	\$42	-\$77	\$3.08 plus 17%
\$50	-\$100	\$4.77 plus 20%	\$77	-\$163	\$9.03 plus 16%
\$100	-\$135	\$14.77 plus 18%	\$163	-\$269	\$22.79 plus 19%
\$135	-\$212	\$21.07 plus 21%	\$269	-\$385	\$42.93 plus 21%
\$212	—	\$37.24 plus 24%	\$385	—	\$67.29 plus 25%

سوف تحسب ضريبة الولاية كواحد في المائة من إجمالى الدخل إذا كان في حدود \$150 أسبوعياً ، وواحد ونصف في المائة لأى دخل إضافى حتى \$500 و 2% لأى زيادة عن \$500 سوف تحسب الضريبة المحلية على أساس 1% من أول \$200 من إجمالى الدخل . الدخل الأسبوعى الذى يزيد عن \$200 لن تحسب عليه ضريبة على المستوى المحلى .

٦ - ٤٩ عدل برنامج ارتداد الكرة في المثال ٦ - ٢٨ لحل المسألة التالية . تحدد وضع الهدف على بعد مسافة L قدم من المصدر

(وتكون قيمة L كمية مدخلة) . قرر بطريقة المحاولة والخطأ ما هي السرعة الأفقية التي يجب أن تأخذها الكرة من أجل تحقيق الهدف بعد الارتداد الثاني . نفرض أن الهدف هو دائرة صغيرة موضوعة على الأرض . بين موضع الهدف على المخرج البياني .

٦ - ٥٠ اكتب برنامج ببسك يولد صورة بيانية على ورق الطباعة لكل من الدوال الآتية :

(أ) $y = 2\sqrt{x}$ لقيم x تتراوح ما بين 0 إلى 10

(ب) $y = x^3$ لقيم x تتراوح ما بين -1 إلى 1

(ج) $y = 2e^{-0.1t} \sin 0.5t$ لقيم t تتراوح ما بين 0 إلى 60 (أنظر المسألة ٥ - ٤٨) .

وفي كل حالة ارسم نقطاً كافية للرسم حتى يرى المنحنى بطريقة واضحة .

٦ - ٥١ اكتب برنامج ببسك يرسم صورة للعلم الأمريكي . استخدم نجمة (*) للإشارة إلى كل نجمة من نجوم العلم . ومثل كل خط من خطوط العلم بعدة سطور مكررة من الحروف R أو W ممتداً في ذلك على لون الخط .

٦ - ٥٢ جهز مخططاً تمهيدياً مفصلاً وخريطة سير عمليات مناظرة ثم برنامج ببسك كامل لكل مسألة من المسائل التالية . ضمن في البرنامج دوال وبرامج فرعية في أي مكان تجده مناسباً .

(أ) احسب متوسط مجموعة من الأرقام عددها N . نفذ الحسابات في البرنامج بواسطة دالة معرفة بواسطة المبرمج . ثم استخدم البرنامج لتشغيل بيانات درجات الحرارة المعطى في المسألة ٥ - ٥٤ .

(ب) طول البرنامج المعطى في المسألة ٦ - ٥٢ (أ) لحساب انحرافات كل نقطة عن المتوسط . استخدم البرنامج لتشغيل بيانات درجات الحرارة المعطى في المسألة ٥ - ٥٤ . هل يمكن لدالة معرفة بواسطة المبرمج أن تستعمل لمثل هذا الفرض .

(ج) احسب المساحة تحت المنحنى ، باستخدام الطرق التي تم وصفها في المسائل ٥ - ٥٧ (د) و ٥ - ٥٧ (ك) . استخدم البرنامج لحساب المساحة تحت المنحنى $y = x^3$ بين الحدود $x = 1$ و $x = 4$.

(د) طريقة أخرى لحساب المساحة تحت منحنى هي استخدام طريقة مونت كارلو ، وهي بالتالي تستخدم لتوليد الأرقام العشوائية . نفرض أن المنحنى $y = f(x)$ منحنى موجب لأى قيمة من قيم x بين الحدين الأدنى والأعلى المعطيين وهما $x = a$ و $x = b$. اجعل أكبر قيمة من قيم y هي y^* وسوف نواصل في طريقة مونت كارلو كالتالى :

(i) إبدأ بعدد قيمته الأولية صفر

(ii) ولد رقم عشوائى ، r_x ، حيث تقع قيمته بين a و b

(iii) احسب قيمة $y(r_x)$

(iv) ولد رقم عشوائى آخر ، r_y ، حيث تقع قيمته بين 0 ، y^*

(v) قارن الرقم r_y بقيمة $y(r_x)$ إذا كانت قيمة r_y أقل من أو تساوى قيمة $y(r_x)$ ، فإن هذه النقطة سوف تقع تحت أو على المنحنى المطلوب . من ثم تزداد قيمة العداد بواحد .

(vi) تكرر الخطوات إبتداء من الخطوة (ii) إلى الخطوة (v) عدداً كبيراً من المرات . وكل مرة من هذه المرات تسمى دورة .

(vii) عند الانتهاء من عدد معين من الدورات ، نحسب العدد الكسرى من النقاط التي تقع تحت المنحنى وتسمى F وتنتج من قسمة قيمة العداد على العدد الإجمالى للدورات . وبذلك نحصل على المساحة تحت المنحنى وهي :

$$A = Fy^* (b - a)$$

اكتب برنامج يبسك لتنفيذ هذا الإجراء . واستخدم البرنامج في إيجاد المساحة تحت المنحنى $y = x^2$ لابين الحدود $x = 0$ و $x = 1$ قرر كم عدد الدورات المطلوبة للحصول على الإجابة الصحيحة بثلاثة أرقام معنوية .
قارن وقت الحاسب المطلوب لهذه المسألة بالوقت المطلوب في المسألة رقم ٥ - ٧ (ك) .

(هـ) احسب متوسط درجات كل طالب في الفصل الدراسي ، ثم احسب متوسط الفصل الدراسي من المتوسطات الفردية .
(أنظر المسائل ٤ - ٨ (ز) إلى ٤ - ٨ (ط)) . حدد القيمة الوسيطة للمتوسطات الفردية (وهي قيمة مساوية أو تتعدد منتصف المتوسطات الفردية) . استخدم دالة لحساب المتوسطات ، ودالة أخرى لحساب الوسيط . ثم طبق البرنامج على البيانات المعطاة في المسألة ٤ - ٨ (ز) . هل يمكن استخدام برنامج فرعية بدلاً من الدوال ؟

(و) بتغير عشوائى موزع طبيعياً x ، متوسطة μ وانحرافه المعيارى σ يمكن توليده من الصيغة التالية :

$$x = \mu + \sigma \frac{\sum_{i=1}^N r_i - N/2}{\sqrt{N/12}}$$

حيث r_i هي رقم عشوائى موزع بانتظام وقيمه تقع بين 0,1 وغالباً ماتختار قيمة $N = 12$ عند استخدام هذه الصيغة .
والجزء الأساسى لهذه الصيغة الذى يفهم ضمناً هو نظرية الحد المركزية ، حيث تنص على أن مجموعة من قيم المتوسطات لمتغيرات عشوائية موزعة توزيعاً منتظماً سوف تكون موزعة طبيعياً .

اكتب برنامج يبسك يولد أرقاماً معطاة لمتغيرات عشوائية موزعة طبيعياً بمتوسطات وانحرافات معيارية معطاة .
اجعل عدد المتغيرات العشوائية والمتوسطات والانحرافات المعيارية كلها معاملات إدخال .

استخدم البرنامج لتوليد هيستوجرام للتوزيع الطبيعى حيث $\mu = 2.5$ و $\sigma = 1.5$

(ز) اكتب برنامج يبسك يسمح لشخص أن يلعب لعبة تيك - تاك - تو ضد الحاسب . اكتب البرنامج بطريقة يمكن أن يكون فيها الحاسب أما اللاعب الأول أو اللاعب الثانى . إذا كان الحاسب هو اللاعب الأول ، فدع الحركة الأولى تولد عشوائياً . اكتب حالة اللعبة كاملة بعد كل حركة . اجعل الحاسب يتعرف على الطرف الفائز عند حدوث ذلك .

(ح) اكتب برنامج يبسك يحاكي لعبة (بلاك جاك) بين لاعبين ولاحظ أن الحاسب لن يشارك في هذه اللعبة كلاعب ولكنه ببساطة سوف يوزع الورق لكل لاعب ويعطى كل لاعب « ضربة » أو أكثر (وذلك يعنى كروتاً إضافية) عند الطلب .

وتوزع الكروت بالترتيب ، أولاً كرت لكل لاعب ثم كرت آخر لكل لاعب ويمكن أن تطلب ضربات إضافية بعد ذلك .

والهدف من اللعبة هو الحصول على 21 نقطة أو أى عدد ممكن من النقاط ولكن لا تتعدى 21 نقطة في كل يد .
ويصبح اللاعب غير مؤهل أوتوماتيكياً إذا تعدت النقاط بيده عن 21 نقطة . وتحسب الصور على أساس 10 نقط ، ويمكن أن يحسب كرت الواحد بنقطة أو 11 نقطة . وبذلك يمكن أن يحصل اللاعب على 21 نقطة من أول كرتين (بلاك جاك !) إذا كان الكرتين اللذين تم توزيعهما أحدهما واحد والأخرى أما عشرة أو صورة . وإذا كانت عدد النقاط التى تم الحصول عليها من أول كرتين قليلة فيمكن أن يطلب اللاعب بعد ذلك ضربة أو أكثر .

يجب أن يستخدم مولد الرقم العشوائى ليحاكى توزيع الكروت . وتأكد من تقسيم شرط عدم توزيع نفس الكرت أكثر من مرة .

(ط) لعبة الروليت تلعب بمجلة تحتوي على 38 مربعا مختلفا حول محيطها . بينهما مربعان ، أرقامهما 00 و 00 ، لونهما أخضر ، 18 مربعا لونها أحمر ، 18 مربعا لونها أسود . وتتعاقد المربعات السوداء والخضراء وترقم من 1 إلى 36 بترتيب عشوائي .

تدار بلية صغيرة بداخل المجلة ، والتي سوف تستقر في آخر الأمر في حفرة وراء أحد المربعات . وتلعب اللعبة بالرهان على نتيجة الدوران ، بأى طريقة من الطرق التالية :

- ١ - بانتقاء مربع واحد أحمر أو أسود ، مقابل 35 إلى 1 (وبذلك إذا راهن اللاعب بدولار واحد وكسب الرهان ، فسوف يتسلم مبلغا إجماليا وقدره \$36.00 - الدولار الأصلي علاوة على \$35.00 إضافي) .
- ٢ - بانتقاء لون (أما أحمر أو أسود) مقابل 1 إلى 1 (وبذلك إذا اختار اللاعب اللون الأحمر وراهن بدولار واحد ، فسوف يتسلم \$2.00 إذا استقرت البلية وراء أى مربع أحمر) .
- ٣ - بانتقاء أما الأرقام الفردية أو الزوجية (باستبعاد 0 و 00) مقابل واحد لواحد .

٤ - بانتقاء أما 18 رقما السفلية أو 18 رقما العليا مقابل واحد لواحد .
وسوف يخسر اللاعب أوتوماتيكيا إذا استقرت البلية وراء أى من المربعات الخضراء (0 أو 00) .
اكتب برنامج يسلك من النوع التخطي الذي يحاكي لعبة الروليت . واسمح للاعب أن ينتقى أى نوع من اللعب حسب رغبته . ثم اطلع نتيجة كل لعبة تتبعها رسالة مناسبة تشير إلى خسارة اللاعب أو مكسبه .

(د) اكتب برنامج يسلك يكود أو يفك شفرة سطر من النص (سلسلة حرفية) ولتكويد السطر اتبع مايل :

- ١ - تحويل أى حرف (حتى الأماكن الحالية) لما يقابله من كود ASCII .
 - ٢ - ولد رقم عشوائي صحيح موجب . أضف هذا الرقم الصحيح إلى قيمة كود ASCII المكافئة لكل حرف . (سوف يستخدم نفس الرقم الصحيح في السطر الكامل) .
 - ٣ - افرض أن N1 تمثل أقل قيمة مسموح بها في كود ASCII و N2 تمثل أعلى قيمة مسموح بها . إذا تعدت قيمة الرقم الذى تم الحصول عليها في الخطوة ٢ العالية قيمة N2 (أى الرقم الأصل المقابل لكود ASCII مضافا إليه ومن الرقم العشوائي) ، فاطرح أكبر رقم مسموح به لمضاعفات N2 من هذا الرقم ، ثم أضف الباقي للرقم N1 ثم فإن الرقم الذى يكود يقع بين N1 و N2 وعلى ذلك تمثل دائما بعض الحروف .
 - ٤ - اطلع الأحرف التى تناظرها حسب كود ASCII .
- سوف يعكس الإجراء في حالة فك شفرة سطر من النص ، ومع ذلك ، تأكد ، أن نفس الرقم العشوائي هو المستخدم في حالة التكويد وفك الشفرة .

(ك) اكتب برنامج يسلك . يحاكي لعبة (بنجو) BINGO اطلع كل توافقية حرف - رقم كما يتم سحبها . تأكد من أن التوافقية لا تسحب إلا مرة واحدة . تذكر أن كلا من الحروف B-I-N-G-O تناظر مدى معين من الأرقام ، كما يل :

B:	1-15
I:	16-30
N:	31-45
G:	46-60
O:	61-75

الفصل ٧

المتجهات والمصفوفات Vectors and Matrices

تعلّمنا في الفصل الخامس أن كل عناصر المجموعة المتراصة يمكن أن يشار إليها جميعاً بإعطاء اسم مجموعة متراصة مشترك . بينما يلزم عند التعامل مع المجموعات المتراصة أن نتعامل مع كل عنصر من عناصر المجموعة المتراصة على حدة (أى كل متغير ذى دليل) . وغالباً ما يتم إنجاز ذلك بواسطة الحلقة التكرارية FOR-TO

تحتوى معظم نسخ البيسك على مجموعة خاصة من الجمل . تعرف باسم جمل المصفوفات . وذلك للقيام بعمليات المجموعات المتراصة الأكثر شيوعاً . وغالباً سوف تستخدم جملة مصفوفات واحدة لعملية معينة . وبذلك يمكن تنفيذ عملية معينة على كل عناصر المجموعة المتراصة بدون استخدام حلقة FOR-TO . في هذا الفصل سوف نرى كيف يمكننا إنجاز ذلك .

٧ - ١ عمليات المتجهات والمصفوفات VECTOR AND MATRIX OPERATIONS

« متجه » و « مصفوفة » مصطلحات رياضية تشير إلى قائمة أو إلى جدول على الترتيب . وبذلك يكون المتجه هو مجموعة متراصة لها بعد واحد والمصفوفة هي مجموعة متراصة ذات بعدين وحيث أن المتجه هو حالة خاصة من المصفوفة ، فإن معظم القواعد العامة التي تطبق على المصفوفات يمكن تطبيقها على المتجهات .

وكما ذكرنا سابقاً ، سوف نستخدم متغيرات ذات أدلة لتمثل كل عنصر من عناصر المجموعة المتراصة على حدة . وفي حالة المصفوفة سوف نجعل الدليل الأول يشير إلى الصف والدليل الثاني يشير إلى العمود . وبذلك فإن $A(3, 2)$ سوف تمثل العنصر في الصف الثالث والعمود الثاني من المصفوفة A . وعلاوة على ذلك سوف نشير إلى مصفوفة لها عدد m من الصفوف و n من الأعمدة كمصفوفة $m \times n$ (تذكر أن حجم المتجه أو المصفوفة يجب أن يحدد بواسطة جملة DIM إذا تعدت قيمة الدليل الرقم 10) .

أغلب عمليات المتجهات والمصفوفات الشائعة هي الجمع والطرح والضرب اللاموجه والضرب الموجه . ولكل من هذه العمليات جملة بيسك خاصة . وتحتوى اللغة أيضاً على جملة تحديد قيم المصفوفة . وسوف تم مناقشة كل جملة من هذه الجمل على حدة فيما بعد .

تحديد قيمة Assignment

جملة تحديد قيم المصفوفة تكون على النحو التالي :

$$10 \text{ MAT } C = A$$

يترتب على هذه الجملة تحديد قيمة لكل عنصر من عناصر المصفوفة C بقيمة العنصر المناظر من المصفوفة A .

مثال ٧-١

نفرض أن المصفوفة A تمثل المصفوفة (2 × 3) التالية :

$$A = \begin{bmatrix} 3 & 5 & -9 \\ 2 & -6 & 7 \end{bmatrix}$$

وجملة المصفوفة

$$10 \text{ MAT } C=A$$

سوف يترتب عليها جعل عناصر المصفوفة (2 × 3) التي تسمى C كالتالي :

$$C = \begin{bmatrix} 3 & 5 & -9 \\ 2 & -6 & 7 \end{bmatrix}$$

تستعمل المتغيرات ذات الأدلة للإشارة إلى كل عنصر من عناصر المصفوفة على حدة وبذلك فإن :

$$C(2, 3) = 7, C(2, 2) = -6, C(2, 1) = 2, C(1, 3) = -9, C(1, 2) = 5, C(1, 1) = 3$$

الجمع Addition

تم عملية جمع المصفوفات مجملة تكتب على النحو التالي :

$$10 \text{ MAT } C=A+B$$

نتيجة هذه الجملة هو تحديد كل عنصر من عناصر المصفوفة C بحاصل جمع العنصرين المناظرين في المصفوفتين A ، B ، أي $C(I, J) = A(I, J) + B(I, J)$. لابد أن تكون المصفوفتان A و B لهما نفس عدد الصفوف ونفس عدد الأعمدة .

مثال ٧-٢

نفرض أن كلا من المصفوفتين A و B (2 × 3) وقيم عناصرها كما يلي :

$$A = \begin{bmatrix} 3 & 5 & -9 \\ 2 & -6 & 7 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 2 & 0 \\ -4 & 5 & 1 \end{bmatrix}$$

وسوف تسبب الجملة :

$$10 \text{ MAT } C=A+B$$

في جعل عناصر المصفوفة (2 × 3) C كما يلي :

$$C = \begin{bmatrix} (3+2) & (5+2) & (-9+0) \\ (2-4) & (-6+5) & (7+1) \end{bmatrix} = \begin{bmatrix} 5 & 7 & -9 \\ -2 & -1 & 8 \end{bmatrix}$$

يمكن أن تعدل القيم المعطاة لمصفوفة ما بجملة الجمع ، مثال لذلك جملة الجمع بالصورة التالية :

$$10 \text{ MAT } A=A+B$$

مسموح بها . بينما جملة الجمع المتعددة مثل :

$$10 \text{ MAT } D=A+B+C$$

غير مسموح بها .

الطرح Subtraction

جملة طرح المصفوفة مشابهة لجملة جمع المصفوفة تماماً فيما عدا استبدال علامة الجمع (+) بعلامة الطرح (-) ، وبذلك فالجملة :

$$10 \text{ MAT } C=A-B$$

سوف تتسبب في إعطاء قيم لعناصر المصفوفة C مساوية لحاصل طرح القيتين المناظرتين في A و B أي $C(I, J)=A(I, J)-B(I, J)$ ويجب أن يكون للمصفوفتين A و B نفس عدد الأعمدة ونفس عدد الصفوف .

مثال ٧ - ٣

نفرض أن A و B مصفوفتان (2×3) ولها نفس العناصر كما في مثال ٧ - ٣ . فبجملة المصفوفة :

$$10 \text{ MAT } C=A-B$$

سوف يترتب عليها إعطاء عناصر المصفوفة C القيم التالية :

$$C = \begin{bmatrix} (3-2) & (5-2) & (-9-0) \\ (2+4) & (-6-5) & (7-1) \end{bmatrix} = \begin{bmatrix} 1 & 3 & -9 \\ 6 & -11 & 6 \end{bmatrix}$$

وكما ذكرنا في جملة جمع المصفوفات ، فيمكن تعديل القيم المعطاة للمصفوفة من خلال جملة الطرح . وبذلك فإن جملة مثل :

$$10 \text{ MAT } A=A-B$$

مسموح بها . في حين أن جملة المصفوفات التالية :

$$10 \text{ MAT } D=A-B-C$$

و

$$10 \text{ MAT } D=A+B-C$$

غير مسموح بهما .

الضرب غير الموجه Scalar Multiplication

في الضرب غير الموجه نضرب كل عنصر من عناصر المصفوفة ، برقم ثابت . ويمكن إنجاز ذلك في البيسك بواسطة الجملة التالية :

$$10 \text{ MAT } C=(K)*A$$

حيث A و C مصفوفتان و K متغير عادي . وسوف يكون كل عنصر من عناصر C له القيمة $C(I, J) = (K)*A(I, J)$

مثال ٧ - ٤

نفرض أن المصفوفة A هي نفس المصفوفة (2×3) المعطاة في الأمثلة ١-٧ ، ٢-٧ ، ٣-٧ وأن K متغير له القيمة 3.5 فإن الجملة :

$$10 \text{ MAT } C=(K)*A$$

سوف يترتب عليها إعطاء عناصر المصفوفة C القيم التالية :

$$C = (3.5) * \begin{bmatrix} 3 & 5 & -9 \\ 2 & -6 & 7 \end{bmatrix} = \begin{bmatrix} 10.5 & 17.5 & -31.5 \\ 7 & -21 & 24.5 \end{bmatrix}$$

لا يستلزم أن يكون الحد الموجود بين القوسين مجرد متغير . ولكن يمكن أن يظهر بين القوسين أيضاً ثوابت أو متغيرات ذات أدلة أو صيغ رياضية أو إشارة إلى دوال . ولكن الشرط هو ؛ أن هذا الحد يجب أن يمثل كمية رقمية . وهذا الحد العددي يجب أن يكون محصوراً بين قوسين .

مثال ٧ - ٥

مبين فيما يلي عدة أمثلة صحيحة لجمع الضرب غير الموجه :

$$\begin{aligned} 10 \text{ MAT } C &= (100) * A \\ 10 \text{ MAT } C &= (2 * X + Y) * A \\ 10 \text{ MAT } C &= (\text{SQR}(P + 2 + Q + 2)) * A \end{aligned}$$

وفي هذه الأمثلة A و B مصفوفتان و X و Y و P و Q متغيرات عددية عادية و SQR تمثل الدالة المكتوبة للجذر التربيعي .

يمكن أن تعدل قيمة المتجه بواسطة جملة الضرب غير الموجه . وبذلك فإن الجملة :

$$10 \text{ MAT } A = (10) * A$$

مسموح بها . وذلك كما حدث في جمع المصفوفة وطرح المصفوفة . بينما جملتين مثل

$$10 \text{ MAT } C = (10) * A * B$$

و

$$10 \text{ MAT } C = (10) * A + B$$

غير مسموح بهما .

ضرب المصفوفات Matrix Multiplication

يمكن ضرب مصفوفتين إذا كان عدد أعمدة المصفوفة الأولى مساوياً لعدد صفوف المصفوفة الثانية . والنتيجة سوف تكون مصفوفة لها نفس عدد صفوف المصفوفة الأولى ونفس عدد أعمدة المصفوفة الثانية . وبذلك فإذا كانت A هي مصفوفة $(k \times m)$ و B مصفوفة $(m \times n)$ فإن جملة المصفوفة $C = A * B$ سوف تولد مصفوفة C لها عدد صفوف مساو k وعدد أعمدة مساو n وكل عنصر من عناصر المصفوفة C سوف نحصل عليه نتيجة للمعملية :

$$C(I,J) = A(I,1) * B(1,J) + A(I,2) * B(2,J) + \dots + A(I,K) * B(K,J)$$

ويمكن القيام بعملية ضرب المصفوفات في البيسك بواسطة الصيغة :

$$10 \text{ MAT } C=A*B$$

حيث كل من A و B و C مصفوفات ، حيث A لها عدد من الأعمدة مساو لعدد الصفوف الموجودة في B .

مثال ٧ - ٦

نفرض أن لدينا المصفوفتين التاليتين :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \quad B = \begin{bmatrix} 9 & 5 & 1 \\ 8 & 4 & 0 \\ 7 & 3 & 9 \\ 6 & 2 & 8 \end{bmatrix}$$

فإن جملة المصفوفة :

$$10 \text{ MAT } C=A*B$$

سوف تنتج مصفوفة $C (2 \times 3)$

$$C = \begin{bmatrix} 70 & 30 & 60 \\ 190 & 86 & 132 \end{bmatrix}$$

حيث يمكن الحصول على كل عنصر من عناصر المصفوفة كما يلي

$$\begin{aligned} C(1,1) &= (1 \times 9) + (2 \times 8) + (3 \times 7) + (4 \times 6) = 70 \\ C(1,2) &= (1 \times 5) + (2 \times 4) + (3 \times 3) + (4 \times 2) = 30 \\ C(1,3) &= (1 \times 1) + (2 \times 0) + (3 \times 9) + (4 \times 8) = 60 \\ C(2,1) &= (5 \times 9) + (6 \times 8) + (7 \times 7) + (8 \times 6) = 190 \\ C(2,2) &= (5 \times 5) + (6 \times 4) + (7 \times 3) + (8 \times 2) = 86 \\ C(2,3) &= (5 \times 1) + (6 \times 0) + (7 \times 9) + (8 \times 8) = 132 \end{aligned}$$

ولكن بعكس جمل المصفوفات التي تم ذكرها سابقاً فإن المصفوفة لا يمكن تعديل قيمها بواسطة جملة ضرب المصفوفة . ولا يمكن أيضاً إجراء عملية الضرب على أكثر من مصفوفتين في وقت واحد . وبذلك فإن الجملتين التاليتين :

$$10 \text{ MAT } A=A*C$$

$$10 \text{ MAT } D=A*B*C$$

غير مسموح بهما ، بينما يمكن ضرب مصفوفة في نفسها ، أى ، يمكن كتابة :

$$10 \text{ MAT } C=A*A$$

بشرط أن تكون المصفوفة A مصفوفة مربعة (أى يجب أن يكون لها عدد من الصفوف مساوياً لعدد الأعمدة) .

٧ - ٢ إدخال / إخراج المتجهات والمصفوفات VECTOR AND MATRIX INPUT/OUTPUT

عمليات إدخال وإخراج المصفوفات يمكن القيام بها بنفس الطريقة التي تم بها العمليات العادية للإدخال / الإخراج . يحددنا البيسك بثلاث جمل (للإدخال والإخراج) I/O من جمل المصفوفات وهي **MAT READ** ، **MAT PRINT** ، و **MAT INPUT** . وسوف نناقش كلا على حدة فيما يلي .

MAT READ

الفرض من جملة MAT READ هو إدخال قيم لعناصر المتجه أو المصفوفة . تستخدم هذه الجملة مقترنة بجملة أو أكثر من جملة DATA (انظر القسم ٥ . ٥) . يمكن أن تظهر جملة MAT READ النموذجية كما يلي :

10 MAT READ A

حيث تمثل A متجهاً أو مصفوفة لها أبعاد سبق تحديدها .

يترتب على تنفيذ جملة MAT READ إعطاء القيم الموجودة في كتلة البيانات لعناصر المتجه أو المصفوفة المناسبة . يبدأ التحديد دائماً بالدليل (أو الأدلة) المساوية للواحد الصحيح ، أى أن العنصر الذى رقه صفر يتم تجاهله . في حالة المصفوفة ، يتم تحديد البيانات على أساس صف ثم الصف التالى .

مثال ٧ - ٧

جزء من برنامج يبسك مبين فيما يلي :

```
10 DIM A(5,3)
...
40 MAT READ A
...
200 DATA 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29
```

ويترتب على تنفيذ هذا البرنامج تحديد عناصر المصفوفة A بالقيم التالية :

A(1,1)=1	A(1,2)=3	A(1,3)=5
A(2,1)=7	A(2,2)=9	A(2,3)=11
A(3,1)=13	A(3,2)=15	A(3,3)=17
A(4,1)=19	A(4,2)=21	A(4,3)=23
A(5,1)=25	A(5,2)=27	A(5,3)=29

لاحظ أن البيانات تم تحديدها صفّاً بصف .

لاحظ أن المصفوفة A تحتوى حقيقة على 24 عنصراً ، حيث أن الأدلة تتراوح ما بين 0 و 5 ومن 0 إلى 3 على الترتيب . ولكن 15 عنصراً فقط من A هى التى تحدد بقيم ، بينما يتم تجاهل العناصر ذات الدليل صفر .

إذا لم تكتب جملة DIM فإن A تتكون أوتوماتيكياً من 121 عنصراً (كل دليل يتراوح من 0 إلى 10) . وبذلك يكون المطلوب 100 قيمة في كتلة البيانات . وسوف يترتب على إعطاء 15 قيمة فقط ، كما في جملة ATA السابقة ، حدوث خطأ .

يمكن أن تحتوى جملة MAT READ واحدة على عدة متجهات ومصفوفات إذا أردنا ذلك . يجب فصل المتجهات والمصفوفات المتعاقبة بواسطة فاصلة (,) وسوف تقرأ كل عناصر المتجه أو المصفوفة الأولى بالكامل قبل أى عنصر من عناصر المتجه أو المصفوفة الثانية و ... وهكذا . وكما ذكرنا سابقاً سوف تحدد قيم المصفوفة صفّاً بصف .

جزء من برنامج ببسك يمثل فيما يلي :

```
10 DIM X(2,2),Y(5),Z(2,3)
50 MAT READ X,Y,Z
150 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
```

وسوف يترتب على تنفيذ هذا البرنامج تحديد عناصر X و Y و Z بالقيم التالية :

X(1,1)=1	Y(1)=5	Z(1,1)=10
X(1,2)=2	Y(2)=6	Z(1,2)=11
X(2,1)=3	Y(3)=7	Z(1,3)=12
X(2,2)=4	Y(4)=8	Z(2,1)=13
	Y(5)=9	Z(2,2)=14
		Z(2,3)=15

وبذلك نرى أن القيم الأربع الأول الموجودة في كتلة البيانات تغطي للمتجه X والقيم الخمس التالية سوف تغطي للمتجه Y والقيم الست الأخيرة للمصفوفة Z . وتحدد قيم المصفوفات صفياً بصف .

MAT PRINT

تستخدم جملة MAT PRINT لطباعة عناصر المتجه أو المصفوفة . يمكن كتابة جملة MAT PRINT نموذجية كما يلي :

```
10 MAT PRINT A
```

حيث تمثل A إما متجهاً أو مصفوفة ، وسوف تطبع عناصر A بشكل عمودي إذا كانت A متجهاً . وبشكل جدول . أو بصورة صف بصف إذا كانت A مصفوفة . وكما في جملة MAT READ سوف يتم تجاهل العناصر التي تحمل الدليل صفراً .

سوف تفصل عناصر كل صف من المصفوفة عن بعضها ، بعد أقصى 5 عناصر في كل سطر مطبوع . من ثم فيمكن أن يكون المطلوب عدة سطور لكل صف . وسوف يطبع سطر خال بين السطور المتعاقبة ، وبذلك نستطيع التمييز بين صف وآخر .

ادرس برنامج البيسك التالي :

```
10 DIM X(3,8),Y(6)
20 MAT READ X,Y
30 MAT PRINT X
40 MAT PRINT Y
50 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
60 DATA 16,17,18,19,20,21,22,23,24,25,26,27,28,29,30
70 END
```

سوف يفتح المخرج التالي عند تنفيذ هذا البرنامج :

1 6	2 7	3 8	4	5
9 14	10 15	11 16	12	13
17 22	18 23	19 24	20	21
25				
26				
27				
28				
29				
30				

لاحظ أن كل صف من X يتطلب سطرين ، حيث يمكن طباعة 5 عناصر فقط في كل سطر . يفصل بين كل صفين متتاليين سطر خال .
ولاحظ كذلك أن Y تطبع على شكل عمودى حيث أنها متجه .

يمكن تعديل التباعد بين عناصر المجموعة المتراصة المتعاقبة بوضع فاصلة (,) أو فاصلة منقوطة (;) بعد ذكر اسم المجموعة المتراصة في جملة MAT PRINT ويتم تداول المتجهات بصورة تختلف عن المصفوفات . وفيما يلي القواعد التي تحكم التباعد بين عناصر المجموعة المتراصة :

١ - المتجهات : Vectors

(أ) إذا تبع اسم المتجه فاصلة (,) تطبع العناصر في شكل صف بدلا من شكل عمودى وسوف تكون المسافات بين العناصر متباعدة (لا يزيد عن 5 عناصر في كل سطر) .

(ب) إذا تبع اسم المتجه فاصلة منقوطة (;) فسوف تطبع العناصر في شكل صف مع ترك أقل مسافة بينها .

٢ - المصفوفات : Matrices

إذا تبع اسم المصفوفة فاصلة (,) فلن يؤثر ذلك في التباعد بين عناصر المصفوفة ولا في شكل الخرج بينما إذا تبع اسم المصفوفة فاصلة منقوطة (;) فسوف تطبع المصفوفة صفاً بصف مع ترك أقل مسافة بين العناصر وبعضها . وسوف يتم فصل الصفوف بسطر خال .

مثال ٧ - ١٠

دعنا ندرس برنامج البيسك المعروض في مثال ٧ - ٩ مرة أخرى . إذا تم تغيير جملة MAT PRINT إلى

```
30 MAT PRINT X,
40 MAT PRINT Y,
```

فسوف يتم توليد الخرج التالى عند تنفيذ البرنامج :

1	2	3	4	5
6	7	8		
9	10	11	12	13
14	15	16		
17	18	19	20	21
22	23	24		
25	26	27	28	29
30				

نرى أن المصفوفة X تم طباعتها بنفس الطريقة السابقة ولكن المتجه Y يظهر الآن على شكل صف .

ومن وجهة نظر أخرى ، نفرض أننا استبدلنا الفاصلات (,) في جملة MAT PRINT بالفاصلات المنقوطة (;) ، أي .
دعنا نكتب :

```
30 MAT PRINT X;
40 MAT PRINT Y;
```

فسوف يظهر المخرج كما يلي :

```
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24

25 26 27 28 29 30
```

وسوف يتم فصل العناصر بأقل مسافة ممكنة وتظهر متقاربة .

يمكن ظهور عدة متجهات ومصفوفات في نفس جملة MAT PRINT إذا تطلب الأمر ذلك . ويجب أن تفصل الأسماء المتعاقبة إما بواسطة فاصلة (,) أو فاصلة منقوطة (;) وسوف يحدد شكل كل متجه أو مصفوفة تبعاً لنوع علامة الفصل التي تتبع اسم المتجه أو المصفوفة .

مثال ٧ - ١١

جملتنا MAT PRINT في مثال ٧ - ٩ يمكن أن نستبدلها بجملة واحدة .

```
30 MAT PRINT X,Y
```

وعند تنفيذ البرنامج فسوف تحدد مسافات المخرج كما هو مبين في مثال ٧ - ٩ .

ومن وجهة نظر أخرى ، نفرض جملة MAT PRINT هذه قد تم تحويلها إلى :

```
30 MAT PRINT X,Y
```

(لاحظ إضافة الفاصلة الأخيرة) وعند تنفيذ هذا البرنامج سوف تولد مجموعة من المخرجات بينها مسافات متباعدة ، وسوف يتم عرض عناصر المتجه في شكل صف كما هو مبين في بداية المثال ٧ - ١٠ .

إذا استبدلت الفاصلة (,) بالفاصلة المنقوطة (.) ، أى .

30 MAT PRINT X;Y;

فإن الخرج سوف يظهر بصورة أكثر تلاصقاً ، كما هو مبين في الجزء التالى من مثال ٧ - ١٠ .

وأخيراً ، يجب أن نشير إلى أن جملة MAT PRINT يمكن أن تحتوى على أسماء متجهات أو مصفوفات فقط . أما الصيغ الرياضية أو مراجع الدوال ، ... إلخ فغير مسموح بها . وبذلك فإن جملة على الصورة :

100 MAT PRINT A+B,C*D,(K)*X

غير مسموح بها .

MAT INPUT

تستخدم جملة MAT INPUT لإدخال عناصر متجه من النهاية الطرفية مباشرة . ويمكن أن تظهر جملة MAT INPUT النموذجية كالتالى :

10 MAT INPUT A

حيث تمثل A اسم متجه . وتسمح معظم نسخ البيسك بظهور اسم متجه واحد فقط في جملة MAT INPUT

عند تنفيذ جملة MAT INPUT سوف تظهر علامة الاستفهام (؟) عند بداية سطر جديد ، مشيرة إلى طلب البيانات . سوف يتوقف تنفيذ البرنامج مؤقتاً أثناء إدخال عناصر المتجه المطلوبة تفصل بينها فاصلات (,) . أول قيمة للبيانات سوف تملأ للعنصر (1) A ، والثانية للعنصر (2) A ... وهكذا (حيث A هو اسم المتجه) . وسوف يتم تجاهل العنصر رقم صفر .

بعد الانتهاء من إدخال آخر عنصر ، يجب على المستخدم أن يضغط على مفتاح RETURN ، وبذلك ينسحب في نقل البيانات إلى الحاسب . بعد ذلك يتم مواصلة تنفيذ البرنامج .

من المهم أن نلاحظ أنه من الممكن إدخال أى عدد من البيانات (على أن يؤخذ في الاعتبار أن عدد قيم البيانات لا تتعدى الحد الأقصى المسموح به لعناصر المتجه ، والمحدد في جملة DIM) . ومن ثم يمكن إدخال مجموعة جزئية فقط من العناصر من خلال جملة MAT INPUT وهذه الخاصية تعتمد على مدى استعمال جملة MAT INPUT في عديد من برامج بيسك .

مثال ٧ - ١٢

يحتوى برنامج بيسك على الجملتين التاليتين :

10 DIM A(100)

...

50 MAT INPUT A

عند مقابلة الجملة رقم 50 أثناء تنفيذ البرنامج ، سوف تطبع علامة استفهام (؟) عند بداية سطر جديد . وسوف يتم تعليق تنفيذ البرنامج مؤقتاً .

نفرض أن السطر التالي من البيانات قد تم إدخاله نتيجة الإجابة على علامة الاستفهام :

?12,-3,17,10,62,-87,49,5,39,9,-7,-22

وسوف يتم نقل البيانات للحاسب بعد الضغط على مفتاح RETURN مسبباً تحديد عناصر المتجه A بالقيم التالية :

A(1)=12	A(5)=62	A(9)=39
A(2)=-3	A(6)=-87	A(10)=9
A(3)=17	A(7)=49	A(11)=-7
A(4)=10	A(8)=5	A(12)=-22

لاحظ أن عنصر المتجه A رقم (0) أى A(0) والعناصر من A(13) إلى A(100) لا تتأثر .

في بعض الأحيان يكون المطلوب قيم بيانات عديدة على سطر واحد من النهاية الطرفية . عند حدوث ذلك يمكن إدخال قيم البيانات على أسطر متتالية . يجب أن تستخدم علامة & لتشير أن سطرًا ثانيًا من البيانات سوف يتم إدخاله . يجب أن تظهر (علامة &) بعد آخر قيمة في كل سطر ما عدا آخر سطر . وسوف تطبع علامة استفهام (؟) عند بداية كل سطر .

مثال ٧ - ١٣

نفرض أن السطرين التاليين من بيانات الإدخال تم طباعتهما استجابة لجملة MAT INPUT المبينة في مثال ٧ - ١٢ :

?3,6,9,12,15,18,21,24,27,30&
?33,36,39,42,45,48,51,54,57,60

علامة الاستفهام في بداية السطر الثاني ، تشير لطلب مزيد من البيانات وقد تم توليدها بواسطة علامة (&) في نهاية السطر الأول . وكان يمكن استكمال هذا الإجراء عند الضرورة (أى ، كان يمكن طباعة (&) في نهاية السطر الثاني ، مولداً طلباً لسطر ثالث من البيانات ، ... وهكذا) .

سوف يستأنف تنفيذ البرنامج بعد الضغط للمرة الثانية لعودة العربة (وذلك بعد السطر الثاني من البيانات) . وسوف يعطى المتجه A ذو المائة عنصر قِيَمَ 20 عنصراً الأول $A(1) = 1$ و $A(2) = 6$ و $A(3) = 9$ و $A(4) = 12$ و $A(5) = 15$ و $A(6) = 18$ و $A(7) = 21$ و $A(8) = 24$ و $A(9) = 27$ و $A(10) = 30$ و $A(11) = 33$ و $A(12) = 36$ و $A(13) = 39$ و $A(14) = 42$ و $A(15) = 45$ و $A(16) = 48$ و $A(17) = 51$ و $A(18) = 54$ و $A(19) = 57$ و $A(20) = 60$ ولن تتأثر باقي عناصر المتجه A .

وفي بعض الأحيان يستحب أن نعرف كم عدد القيم التي تم إدخالها من النهاية الطرفية . تسمح الدالة المكتوبة NUM بالإجابة على هذا السؤال . فأيها المستخدم NUM كرجع للدالة سوف تعطينا عدد قيم البيانات التي تم إدخالها في جملة MAT INPUT الأكثر حداثة . لا تتطلب هذه الدالة أى خلاصات .

مثال ٧ - ١٤

مبين فيما يلي جزء من برنامج بيسك :

```
10 DIM A(100)
...
50 MAT INPUT A
60 LET A(0)=NUM
...
150 PRINT "THE LIST CONTAINS"; A(0); " VALUES"
```

عند تنفيذ هذا البرنامج سوف يتم إدخال عدد غير محدد من قيم البيانات ويتم إعطاؤها لعناصر المتجه A. تتسبب جملة 60 في تحديد عدد قيم البيانات وتخزينها في المتغير رقم صفر من المتجه A. ويرتب على تنفيذ الجملة رقم 150 طباعة رسالة تشير إلى الحجم الحقيقي للمتجه A.

نفرض مثلاً ، أنه تم إعطاء 20 قيمة للمتجه A (كما في مثال ٧ - ١٢) فإن الجملة سوف تولد الرسالة التالية :

THE LIST CONTAINS 20 VALUES

تحتوي القائمة على 20 قيمة

حيث تم تحديد قيمة 20 للمتغير رقم (0) A في الجملة 60 .

يمكن أن تستخدم جملة MAT INPUT في بعض نسخ البيسك لإدخال قيم عناصر مصفوفة تماماً كعناصر المتجه . ولكن في المصفوفة يجب أن تحدد العناصر التي يجب إدخالها من خلال جملة MAT INPUT . (وسوف نرى كيف يتم إنجاز ذلك في قسم ٧-٤) . وبذلك تصبح جملة MAT INPUT أقل نفعاً عند إدخال عناصر المصفوفة عنها للمتجه ، حيث لا يمكن إدخال مجموعة جزئية من عناصر المصفوفة . ولهذا السبب ، ولأن معظم نسخ بيسك لا تسمح إطلاقاً باستخدام جملة MAT INPUT مع المصفوفة ، فسوف لا نناقش هذا الموضوع بعد ذلك .

يوضح المثال التالي استخدام عدة جمل مصفوفات في برنامج بيسك كامل من بينها MAT READ و MAT INPUT وسوف نرى برنامج بيسك كاملاً يستخدم جملة MAT INPUT في جزء لاحق من هذا الفصل ، في المثال ٧ - ٢٢ .

مثال ٧ - ١٥ : متاولة المصفوفة Matrix Manipulation

نفرض أن المصفوفتين A و B كل منهما (3 × 3) وتأخذ عناصرها القيم التالية :

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 13 & 15 & 17 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

ونرغب في حساب قيمة صيغة المصفوفة الرياضية التالية :

$$F=5*(A+B)*(A-B)$$

يمكن إنجاز ذلك بسهولة بواسطة جمل المصفوفة التي تم عرضها قبل ذلك في هذا الفصل . (ويمكن حل هذه المسألة أيضاً بدون استخدام جمل المصفوفات ، بالرغم أن البرنامج سوف يصبح أكثر تعقيداً) .

طريقة إجراء الحسابات Computational Procedure

حيث أن الصيغة المعطاة لا يمكن حسابها بجملة مصفوفة واحدة ، فيجب تكوين عدد متتالي من عمليات المصفوفات البسيطة والتي سوف تعطينا النتيجة المطلوبة . ويمكن إنجاز ذلك كالتالي :

$$\begin{aligned} C &= A+B \\ D &= A-B \\ E &= C*D \\ F &= (5)*E \end{aligned}$$

يمكن القيام بكل عملية من العمليات السابقة بجملة مصفوفة واحدة .

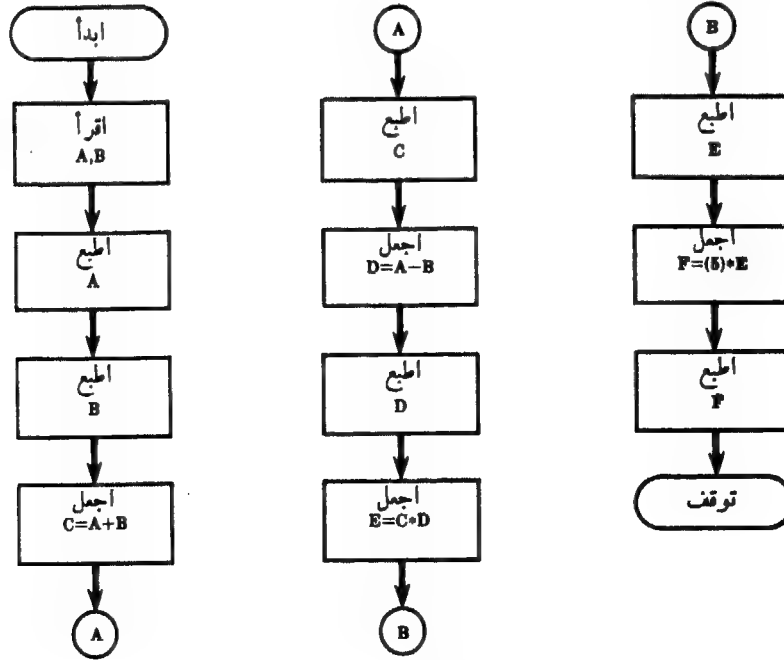
الخطى التمهيدى للبرنامج The Program Outline

هنا نطبع عناصر كل مصفوفة فوراً بعد قراءتها أو حسابها . ويسمح لنا ذلك بملاحظة نتائج كل الخطوات في إجراء الحسابات . وسوف تكون أيضاً نتائج كل مصفوفة ظاهرة تماماً .

وسوف نواصل الحسابات كالتالى :

١ - اقرأ عناصر المصفوفة A و B

٢ - اطبع عناصر المصفوفة A تليها عناصر المصفوفة B



شكل ٧ - ١

٣ - احسب عناصر المصفوفة C.

٤ - اطبع عناصر المصفوفة C.

٥ - احسب عناصر المصفوفة D.

٦ - اطبع عناصر المصفوفة D.

٧ - احسب عناصر المصفوفة E.

٨ - اطبع عناصر المصفوفة E.

٩ - احسب عناصر المصفوفة F.

١٠ - اطبع عناصر المصفوفة F.

١١ - توقف

خريطة سير العمليات المناظرة لذلك مبينة في شكل ٧ - ١

برنامج بيسك The BASIC Program

نرى في شكل ٧ - ٢ برنامج بيسك كامل وذلك لإجراء الحسابات . لاحظ أن البرنامج يتضمن عدة جمل مختلفة . من جمل المصفوفات . لاحظ أيضاً أن البرنامج يتضمن جملة DIM ، رغم أن وجودها ليس من الأهمية حيث أن هذا البرنامج يتطلب مصفوفات 3×3 فقط .

```

10 REM EVALUATION OF THE MATRIX FORMULA F=(5)*(A+B)*(A-B)
20 DIM A(3,3),B(3,3),C(3,3),D(3,3),E(3,3),F(3,3)
30 MAT READ A,B
40 PRINT "THE A-MATRIX IS:"
50 MAT PRINT A
60 PRINT "THE B-MATRIX IS:"
70 MAT PRINT B
80 MAT C=A+B
90 PRINT "THE C-MATRIX IS:"
100 MAT PRINT C
110 MAT D=A-B
120 PRINT "THE D-MATRIX IS:"
130 MAT PRINT D
140 MAT E=C*D
150 PRINT "THE E-MATRIX IS:"
160 MAT PRINT E
170 MAT F=(5)*E
180 PRINT "THE F-MATRIX IS:"
190 MAT PRINT F
200 DATA 1,3,5,7,9,11,13,15,17,2,4,6,8,10,12,14,16,18
210 END

```

شكل ٧ - ٢

يحتوى شكل ٧ - ٢ على الخرج الذى تم توليده بواسطة هذا البرنامج ثم طباعة عناصر كل مصفوفة في هذا البرنامج على بعد مسافات متقاربة وفي شكل جدول . وسوف نرى النتائج المطلوبة (العناصر المحسوبة للمصفوفة F) عند نهاية الشكل .

وأخيراً فإننا نوجه نظرك مرة أخرى إلى أن جمل المصفوفات ليست أساسية وكان يمكن إنجاز نفس الشئ بتضمين عدة حلقات تكرارية FOR-TO في برنامجنا . (برنامج مكتوب بهذه الطريقة مبين في مثال ٥ - ١٥) . إلا أن استخدام جمل المصفوفة تبسط البرمجة إلى حد بعيد .

٣ - ٧ مصفوفات خاصة SPECIAL MATRICES

وأحياناً يجب استخدام بعض المصفوفات الخاصة عند القيام بعمليات المصفوفات وذلك مثل مصفوفة الوحدة أو المصفوفة الملبورة أو مقلوب المصفوفة . يحتوى اليبسك على عدد من جمل المصفوفات التى تسمح بتكوين هذه المصفوفات الخاصة . دعنا ندرس كلاماً من هذه الجمل على حدة .

MAT ZER

تستخدم جملة MAT ZER لتحديد عناصر مصفوفة معينة بصفر . ويمكن أن تظهر جملة MAT ZER النموذجية كالتالى :

```
10 MAT A=ZER
```

حيث تمثل A مصفوفة لها أبعاد محددة .

THE A-MATRIX IS:

1 3 5
7 9 11
13 15 17

THE B-MATRIX IS:

2 4 6
8 10 12
14 16 18

THE C-MATRIX IS:

3 7 11
15 19 23
27 31 35

THE D-MATRIX IS:

-1 -1 -1
-1 -1 -1
-1 -1 -1

THE E-MATRIX IS:

-21 -21 -21
-57 -57 -57
-93 -93 -93

THE F-MATRIX IS:

-105 -105 -105
-285 -285 -285
-465 -465 -465

شكل ٧ - ٣

MAT CON

والغرض من هذه الجملة هو تحديد كل عناصر مصفوفة معينة بالواحد الصحيح . ويمكن أن تكتب جملة MAT CON تماماً كالتالي :

10 MAT B=CON

حيث تمثل B مصفوفة لها أبعاد محددة .

MAT IDN

يترتب على هذه الجملة تحديد كل عناصر مجموعة متراصة مربعة بأصفار (0's) ماعدا عناصر القطر الأساسي (أى القطر الذي يتغير من أعلى اليسار إلى أسفل اليمين) حيث يتم تحديدها بالواحد الصحيح (1's) . وتعرف هذه المصفوفة التي تم تحديد عناصرها بهذه الطريقة بمصفوفة الوحدة .

ويمكن كتابة MAT IDN تماماً كالتالى :

10 MAT C=IDN

حيث تمثل المصفوفة C مصفوفة مربعة لها أبعاد محددة . (لاحظ أن المصفوفة يجب أن يكون لها نفس العدد من الأعمدة والصفوف لكي تكون مصفوفة مربعة) .

مصفوفة الوحدة لها الخصائص المهمة التالية : إذا ضربت مصفوفة مربعة D في مصفوفة الوحدة C ، فإن حاصل الضرب سوف يكون ببساطة هو المصفوفة D بمعنى آخر :

$$C*D=D*C=D$$

(مع ملاحظة أن C و D تمثل وفق قواعد ضرب المصفوفات) . ومن ثم فإننا نرى أن ضرب المصفوفات التى تتضمن مصفوفة الوحدة مماثل للضرب العادى حيث أحد العوامل هو الثابت 1 .

مثال ٧ - ١٦

مبين فيما يلى جزء من برنامج بيك :

```
10 DIM A(2,3),B(4,2),C(3,3)
...
70 MAT A=ZER
80 MAT B=CON
90 MAT C=IDN
```

سوف يسبب تنفيذ هذا البرنامج تحديد المصفوفات A و B و C بالقيم التالية :

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

MAT TRN

تسبب جملة MAT TRN توليد صفوف وأعمدة مصفوفة ممتدة (أى تبديلها) وسوف تظهر الجملة كإلى :

10 MAT B=TRN(A)

وبذلك فإنه إذا كانت A مصفوفة أبعادها (m × n) فإن B سوف تكون أبعادها (n × m) وتحدد عناصرها كالتالى :

$$B(I,J)=A(J,I)$$

تسمى المصفوفة B بالمصفوفة المدورة A

ادرس برنامج البليك التالى الذى يحتوى على الجمل التالية :

```
10 DIM A(2,3),B(3,2)
20 MAT READ A
...
50 MAT B=TRN(A)
...
80 DATA 1,3,5,7,9,11
```

الجملة رقم 20 فى تحديد عناصر المصفوفة A بالقيم الموضحة فيما بعد :

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{bmatrix}$$

وتحدد الجملة رقم 50 عناصر المصفوفة B بالقيم الموضحة فيما بعد :

$$B = \begin{bmatrix} 1 & 7 \\ 3 & 9 \\ 5 & 11 \end{bmatrix}$$

لاحظ أن المصفوفة B لها 3 صفوف وعمودين بينما المصفوفة A لها صفين و 3 أعمدة .

MAT INV

مقلوب المصفوفة المربعة هو أيضاً مصفوفة مربعة ولها الخاصية المهمة التالية : حاصل ضرب المصفوفة ومقلوبها مساو لمصفوفة الوحدة . وبمعنى آخر . فإذا كانت A مصفوفة مربعة و B مقلوب هذه المصفوفة ، من ثم :

$$A*B=B*A=C$$

حيث C هي مصفوفة الوحدة ، يجب أن تكون المصفوفة مربعة حتى يمكن حساب مقلوبها . بعض المصفوفات المربعة ، لا يمكن حساب مقلوبها . وعلى ذلك حساب مقلوب مصفوفة مربعة يكون فى بعض الحالات ممكناً وفى حالات أخرى غير ممكن .

يمكن حساب مقلوب المصفوفة (إذا كان ذلك ممكناً) بواسطة جملة MAT INV . ويمكن أن تظهر هذه الجملة كالتالى :

```
10 MAT B=INV(A)
```

حيث A و B مصفوفات مربعة لها أبعاد محددة .

يمكن الحصول على محدد المصفوفة الأصلية فور حساب مقلوب المصفوفة بواسطة الدالة المكتوبة DET وترجع هذه الدالة بقيمة عددية ولا تتطلب أى خلاصة .

من بين أشياء كثيرة أخرى ، يمكن استخدام دالة DET لتحديد ما إذا كانت المصفوفة لها مقلوب أم لا ، حيث تكون قيمة المحدد صفراً فى حالة عدم إمكانية الحصول على مقلوب المصفوفة . ومع ذلك ، تذكر أنه يمكن الرجوع إلى الدالة DET فقط بعد جملة MAT INV إذا أعطيت دالة DET القيمة صفراً لمصفوفة معينة فإن مقلوب المصفوفة التى تم الحصول عليها من جملة MAT INV

السابقة ليس لها أى معنى . وعلاوة على ذلك يمكن أن يكون حساب كل من المقلوب والمحدد غير صحيحة تحت شروط معينة ، على سبيل المثال ، إذا كانت المصفوفة الرئيسية كبيرة (عدد كبير من الصفوف وعدد كبير من الأعمدة) ، أو إذا كانت مفردة تقريباً (أى أن القيمة الحقيقية لمحددها قريبة جداً من الصفر) .

مثال ٧ - ١٨ مقلوب المصفوفة Matrix Inversion

فيما يلي برنامج بسيط يحسب مقلوب مصفوفة ، ويحدد المصفوفة وحاصل ضرب المصفوفة فى مقلوبها .

```
10 DIM A(3,3),B(3,3),C(3,3)
20 MAT READ A
30 MAT B=INV(A)
40 MAT C=A*B
50 MAT PRINT A,B,C
60 PRINT "DETERMINANT=";DET
70 DATA 5,3,1,3,7,4,1,4,9
80 END
```

وسوف ينتج عند تنفيذ هذا البرنامج ادخال القيم التالية لعناصر المصفوفات A و B و C

$$A = \begin{bmatrix} 5 & 3 & 1 \\ 3 & 7 & 4 \\ 1 & 4 & 9 \end{bmatrix} \quad B = \begin{bmatrix} 0.274854 & -0.134503 & 0.0292398 \\ -0.134503 & 0.25731 & -0.0994152 \\ 0.0292398 & -0.0994152 & 0.152047 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

وأيضاً يكون محدد المصفوفة A مساوياً 171

مبين فيما يلي المخرج الناتج من هذا البرنامج :

```
5          3          1
3          7          4
1          4          9

0.274854   -0.134503   2.92398E-2
-0.134503   0.25731    -9.94152E-2
2.92398E-2  -9.94152E-2  0.152047

1.          8.38190E-9   3.72529E-9
5.58794E-9  1.          7.45058E-9
0           7.45058E-9   1.

DETERMINANT= 171
```

لاحظ أن بعض عناصر المصفوفة C صحيح بالتقريب فقط ، وذلك نتيجة الأخطاء التى تحدث عند حسابات عناصر C . وتحدث هذه الأخطاء حين حساب الفرق بين الأرقام المتقاربة جداً فى القيمة .

حل المعادلة الآتية Solution of Simultaneous Equations

تسمح لنا جملة MAT INV لحل نظام من المعادلات الجبرية الخطية الآتية ببساطة شديدة . ولتفهم كيفية عمل الطريقة فسوف نلتفت بمفاتيح كل من مقلوب المصفوفة ومصفوفة الوحدة . ولنفرض مثلاً أننا أعطينا مجموعة n من المعادلات :

والأهمية من هذه النتيجة هو التالي . إن حل نظام معادلات جبريه - خطية آنية مساو لحاصل ضرب مقلوب معاملات المصفوفة ومتجه الطرف الأيمن . يوضح مثال ٧ - ١٩ كيف يمكن تحويل هذه الفكرة ببساطة داخل برنامج بيسك .

مثال ٧ - ١٩ المعادلات الآنية Simultaneous Equations

نفرض أننا أعطينا النظام التالي المكون من 5 معادلات في 5 مجاهيل :

$$\begin{aligned} 11x_1 + 3x_2 + x_4 + 2x_5 &= 51 \\ 4x_2 + 2x_3 + x_5 &= 15 \\ 3x_1 + 2x_2 + 7x_3 + x_4 &= 15 \\ 4x_1 + 4x_3 + 10x_4 + x_5 &= 20 \\ 2x_1 + 5x_2 + x_3 + 3x_4 + 13x_5 &= 92 \end{aligned}$$

ونرغب في تحديد قيم المجاهيل x_1 و x_2 و x_3 و x_4 و x_5

طريقة إجراء الحسابات Computational Procedure

دعنا نعد كتابة المعادلات على صورة مصفوفة كالتالي :

$$C \cdot X = D$$

حيث

$$C = \begin{bmatrix} 11 & 3 & 0 & 1 & 2 \\ 0 & 4 & 2 & 0 & 1 \\ 3 & 2 & 7 & 1 & 0 \\ 4 & 0 & 4 & 10 & 1 \\ 2 & 5 & 1 & 3 & 13 \end{bmatrix} \quad D = \begin{bmatrix} 51 \\ 15 \\ 15 \\ 20 \\ 92 \end{bmatrix}$$

وسوف نحتوي X القيم x_1 و x_2 و x_3 و x_4 و x_5 . ويمكن أن نحصل على هذه القيم ببساطة بواسطة حساب حاصل ضرب المصفوفة :

$$X = E \cdot D$$

حيث E هي مقلوب المصفوفة C .

وفور تحديد قيم x_1 إلى x_5 يمكن اختبار مدى دقة الحل بحساب حاصل الضرب :

$$F = C \cdot X$$

فإذا تم تحديد قيم X بصورة صحيحة ، فإن المتجه F سوف يكافئ المتجه المحدد D تماماً وأى خلاف في القيمة بين عناصر F والعناصر المناظرة في D يعطينا مقياس الأخطاء المتضمن في حساب X :

المخطط التمهيدى البرنامج The Program Outline

يمكن القيام بالحساب كما يلي :

١ - اقرأ عناصر المصفوفة C والمصفوفة D .

٢ - اطبع عناصر المصفوفة C والمصفوفة D .

٣ - احسب عناصر E (وهي مقلوب المصفوفة C) .

٤ - حدد قيم X وذلك بتكوين حاصل الضرب $E \cdot D$

٥ - اطبع عناصر X .

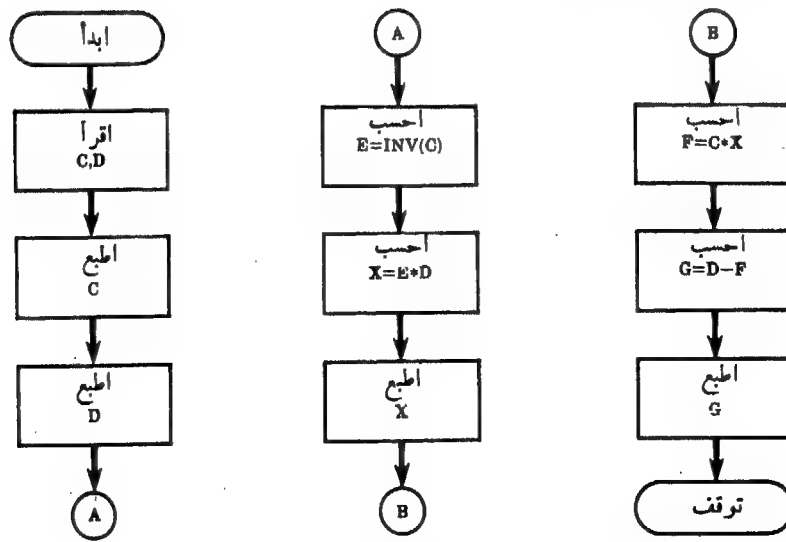
٦ - حدد قيم F وذلك بتكوين حاصل الضرب $C \cdot X$

٧ - احسب متجه الأخطاء $G = D - F$

٨ - اطبع عناصر G .

٩ - توقف .

خريطة سير العمليات المناظرة لذلك مبينة في شكل ٧ - ٤ .



شكل ٧ - ٤

برنامج البيسك The BASIC Program

يبين شكل ٧ - ٥ برنامج بيسك . وهيكل البرنامج مباشر تماماً ، حيث لا يتطلب هذا البرنامج التفرعات ولا الحلقات التكرارية . وهذه الطريقة المبسطة ناجمة عن استعمال جمل المصفوفات ، والتي تبرز المبرمج من تناول تفاصيل المنطق عند التعامل مع المصفوفات .

```

10 REM SOLUTION OF SIMULTANEOUS LINEAR ALGEBRAIC EQUATIONS
20 DIM C(5,5),D(5),E(5,5),F(5),G(5),X(5)
30 MAT READ C,D
40 PRINT "COEFFICIENT MATRIX:"
50 MAT PRINT C
60 PRINT "RIGHT HAND SIDE:"
70 MAT PRINT D
80 MAT E=INV(C)
90 MAT X=E*D
100 PRINT "SOLUTION VECTOR:"
110 MAT PRINT X
120 MAT F=C*X
130 MAT G=D-F
140 PRINT "ERROR VECTOR:"
150 MAT PRINT G
160 DATA 11,3,0,1,2,0,4,2,0,1,3,2,7,1,0
170 DATA 4,0,4,10,1,2,5,1,3,13,51,15,15,20,92
180 END

```

شكل ٧ - ٥

نرى في شكل ٧-٦ المخرج الناتج من البرنامج ، وهو يمدنا بحل 5 معادلات في 5 مجاهيل . والحل المطلوب بالتقريب هو
 $x_1 = 3.0$ ، $x_2 = 2.2$ ، $x_3 = 0.21$ ، $x_4 = 0.15$ ، $x_5 = 5.7$. لاحظ أيضاً أن عناصر متجه الأخطاء G كلها أصغر
 في القيمة من 10^{-6} وهذا يؤكد لنا أن الحل صحيح ومعقول .

٧ - ٤ تغيير أبعاد المجموعات المتراسة CHANGING DIMENSIONS

لقد رأينا أن جملة MAT INPUT تسمح لنا بإدخال عدد غير محدد لعناصر متجه من النهاية للطرفية المركزية ، وبذلك يمدنا
 بخاصية البعد المتغير . وبالإضافة إلى ذلك ، فإن عديداً من جمل المصفوفات تسمح لنا بتغيير أبعاد المجموعة المتراسة أثناء تنفيذ
 البرنامج ، في الحقيقة أننا قادرون على ذلك ، إذا تطلب الأمر ، وهو تغيير حجم المجموعة المتراسة في أماكن عديدة - بداخل البرنامج .
 ويمكن استخدام هذه الإمكانية لتعميم العديد من برامج البيسك التي تتضمن متجهات ومصفوفات .

مصفوفة المعاملات				
11	3	0	1	2
0	4	2	0	1
3	2	7	1	0
4	0	4	10	1
2	5	1	3	13
الطرف الأيمن				
51	15	15	20	92
متجه الحل				
2.97917	2.2156	0.211254	0.152317	5.71503
متجه الخطأ				
-9.53674E-7	-2.38419E-7	-4.76837E-7	-7.15256E-7	-1.90735E-6

شكل ٧-٦

جمل المصفوفات التي تسمح بإعادة تحديد حجم المصفوفة هي : MAT IDN و MAT CON و MAT ZER و MAT READ (لاحظ أن جملة MAT PRINT غير متضمنة) . تنفذ هذه الخاصية بإعطاء الحجم الحقيقي محصور بين قوسين بعد اسم المجموعة المتراسة . وفي حالة المصفوفة يجب أن تفصل بين البعدين بواسطة فاصلة (,) .

مثال ٧-٢٠

يحتوي برنامج بيسك على الجمل التالية :

```
10 DIM A(24,24),B(24,24)
...
50 MAT A=IDN(20,20)
60 MAT B=CON(8,12)
```

عند تنفيذ هذا البرنامج ، سوف يتم حجز عدد 625 عنصراً لكل من المصفوفتين A و B (متضمنة الصف رقم صفر والعمود رقم صفر) . بينما ، الجملة رقم 50 سوف تحدد A على أنها مصفوفة الوحدة وحجمها (20 × 20) وتعطى الجملة رقم 60 عناصر المصفوفة B كلها القيمة 1 وحجم B هو (8 × 12) .

يمكن أن نستخدم إما رقم ثابت أو رقم متغير في تمثيل الأبعاد الحقيقية للمجموعة المتراسة . بينما يجب أن تكون الأبعاد قيم صحيحة موجبة ولا يمكن أن تتعدى الرقم الأقصى لأحجام المجموعة المتراسة في جملة DIM . (أما إذا كانت جملة DIM غير موجودة فإن البعد الحقيقي لا يمكن أن يتعدى الرقم 10) .

مثال ٧ - ٢١

مبين فيما يلي جزء من برنامج بيسك

```
10 DIM P(35),Q(50,50),R(50,50)
...
50 INPUT M,N
...
80 MAT READ P(M),Q(N,N)
90 MAT R=INV(Q)
...
200 DATA...
```

سوف تتسبب الجملة رقم 50 في إدخال قيم M و N (الحجم الحقيقي للمجموعة المتراسة) من النهاية الطرفية المركزية . يمكن إدخال أى أرقام صحيحة موجبة ، أخذين في الاعتبار ألا تتعدى القيم 35 و 50 على الترتيب .

حين مقابلة الجملة رقم 80 فإن عدد M الأول من القيم الموجودة في كتلة البيانات سوف تعطى لعناصر المتجه P ، ثم بعد ذلك عدد (N × N) من القيم التالية (أى 2 ↑ N) سوف تعطى لعناصر المصفوفة Q . في الجملة رقم 90 سوف يعاد تحديد حجم المصفوفة R ضمناً وذلك بعملها (N × N) حيث أن مقلوب المصفوفة يجب أن يكون له نفس حجم المصفوفة المعطاة .

وأخيراً ، يجب أن تكون الجملة رقم 200 موجودة وذلك لتعريف كتلة البيانات (ويمكن أن تكون عدة جمل DATA حقيقة) . لاحظ أن كتلة البيانات يجب أن تحتوى على الأقل قيم بيانات عددها $N \uparrow 2 + M$ (وسوف تهمل أى بيانات زائدة) . أيضاً ، لاحظ أن قيم معينة من كتلة البيانات يمكن أن تعطى لعناصر المتجه P ، أو عناصر المصفوفة Q أولاً تقرأاً ثانياً وذلك يتوقف على القيم المعطاة للمتغيرين M و N .

سوف نرى في المثال التالى أن تعميم برنامج بيسك يمكن أن يحسن وذلك باستخدام خاصية البعد المتغير .

مثال ٧ - ٢٢ توفيق المنحنى بالمربعات الصغرى (لعبة سوق الأوراق المالية)

Least Squares Curve Fitting (Playing the Stock Market)

في هذا المثال سوف نكتب برنامج بيسك لتوافق منحنى مجموعة من البيانات باستخدام طريقة المربعات الصغرى . وسوف نطبق هذا البرنامج بعد ذلك على مسألة توفيق « منحنى الاتجاه » المناسب لمجموعة بيانات الأرباح الميينة في جدول ٧ - ١ لشركة وهمية (تعرف باسم Federated Mousetraps, Incorporated) وبمجرد حصولنا على « منحنى الاتجاه » سوف يكون من الممكن تحديد الأرباح لكل سهم في وقت لاحق مثلاً سنة 1978 . وقياساً على القيمة المتوسطة للنسبة بين السعر والربح ، يمكن تقدير سعر السهم في سنة 1978 ، باستخدام الصيغة الرياضية :

$$P = R \cdot E$$

$P =$ السعر التقديرى لسهم واحد من الأوراق المالية في سنة 1978

حيث

$R =$ القيمة المتوسطة للنسبة بين السعر والربح .

$E =$ الأرباح التقديرية ، دولار لكل سهم ، في سنة 1978

وسوف تستخدم هذه المعلومات كدليل لتقرير هل يمكن شراء أسهم الشركة بالسعر الحالي أم لا .

جدول ٧ - ١ الأرباح السنوية لكل سهم لشركة

Federated Mousetraps, Inc.

السنة	الأرباح . دولار للسهم
1957	\$0.01
1958	0.02
1959	0.02
1960	0.03
1961	0.03
1962	0.04
1963	0.04
1964	0.09
1965	0.24
1966	0.38
1967	0.63
1968	0.93
1969	1.24
1970	1.48
1971	1.73
1972	2.07
1973	2.50
1974	3.12
1975	3.48

طريقة المربعات الصغرى The Least Squares Technique

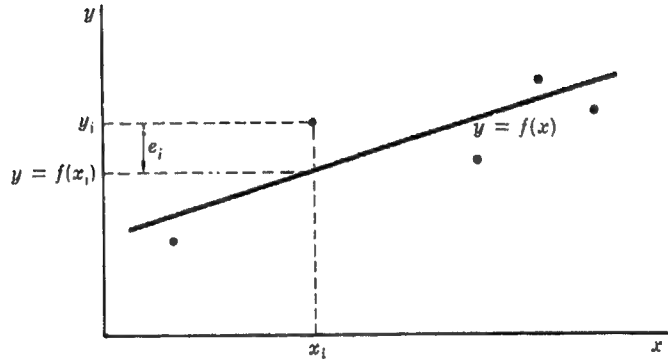
طريقة المربعات الصغرى هي طريقة شائعة لتوفيق منحنى :

$$y = f(x)$$

لمجموعة من نقاط البيانات (y_1, x_1) و (y_2, x_2) و و (y_M, x_M) . والطريقة مبنية على مفهوم تقليل مجموع مربعات الأخطاء :

$$e_1^2 + e_2^2 + \dots + e_M^2$$

حيث e_i هو الخطأ رقم i وهو لى قيمتين معطائين e_i و x_i ؛ الفرق بين نقطة البيانات y_i وقيمة $y = f(x_i)$ والتي تقرأ من المنحنى (انظر شكل ٧ - ٧) .



شكل ٧ - ٧

والطريقة شائعة التطبيق على الدوال الأسية والدوال المرفوعة لقوة معينة ولكثيرة الحدود . وفي كل حالة ، تتطلب الطريقة حل مجموعة من المعادلات الجبرية الخطية الآتية ، حيث الكميات المجهولة هي الكميات الثابتة لمعادلة المنحنى .
نفرض أننا نرغب في أن تمرر الدالة ذات القوة .

$$y = ax^b$$

خلال مجموعة من نقاط البيانات M ولعمل ذلك يجب أن نحل المعادلتين التاليتين لكل من a و b

$$M \log a + \left\{ \sum_{i=1}^M \log x_i \right\} b = \sum_{i=1}^M \log y_i$$

$$\left\{ \sum_{i=1}^M \log x_i \right\} \log a + \left\{ \sum_{i=1}^M (\log x_i)^2 \right\} b = \sum_{i=1}^M (\log x_i)(\log y_i)$$

حيث \sum تشير إلى مجموع (فمثلا $\sum_{i=1}^M \log x_i = \log x_1 + \log x_2 + \dots + \log x_M$) . لاحظ أن هذه المعادلات جبرية وخطية بدلالة المجهولين $\log a$ و b ، وبمجرد حساب قيمة $\log a$ يكون من السهل الحصول على قيمة الثابت a .

والآن نفرض أننا نرغب في تمرير المنحنى الأسى :

$$y = ae^{bx}$$

خلال مجموعة M من نقاط البيانات . والمعادلات المطلوب حلها هي :

$$M \log a + \left\{ \sum_{i=1}^M x_i \right\} b = \sum_{i=1}^M \log y_i$$

$$\left\{ \sum_{i=1}^M x_i \right\} \log a + \left\{ \sum_{i=1}^M x_i^2 \right\} b = \sum_{i=1}^M x_i \log y_i$$

ومرة أخرى فإننا بصدد حل معادلتين جبريتين خطيتين آتيتين لكل من $\log a$ و b .

وإذا كان المنحنى هو كثيرة الحدود :

$$y = c_1 + c_2x + c_3x^2 + \dots + c_{n+1}x^n$$

فإننا نحدد المعاملات c_1 و c_2 و c_3 و \dots و c_{n+1} وذلك بحل المعادلات .

$$Mc_1 + \left\{ \sum_{i=1}^M x_i \right\} c_2 + \left\{ \sum_{i=1}^M x_i^2 \right\} c_3 + \dots + \left\{ \sum_{i=1}^M x_i^n \right\} c_{n+1} = \sum_{i=1}^M y_i$$

$$\left\{ \sum_{i=1}^M x_i \right\} c_1 + \left\{ \sum_{i=1}^M x_i^2 \right\} c_2 + \left\{ \sum_{i=1}^M x_i^3 \right\} c_3 + \dots + \left\{ \sum_{i=1}^M x_i^{n+1} \right\} c_{n+1} = \sum_{i=1}^M x_i y_i$$

.....

$$\left\{ \sum_{i=1}^M x_i^n \right\} c_1 + \left\{ \sum_{i=1}^M x_i^{n+1} \right\} c_2 + \left\{ \sum_{i=1}^M x_i^{n+2} \right\} c_3 + \dots + \left\{ \sum_{i=1}^M x_i^{2n} \right\} c_{n+1} = \sum_{i=1}^M x_i^n y_i$$

وحيث أن الحالتين السابقتين تتضمنان حل معادلات جبرية خطية آتية فإننا سوف ندخل عدة جمل مصفوفات في برنامجنا ، بطريقة مشابهة لما سبق ذكره في المثال ٧ - ١٩ .

تصميم البرنامج Design of the Program

عند تصميم البرنامج سوف ندخل عدداً من الخصائص التي سوف تجعل البرنامج عام وبصورة أوسع وأشمل ، فمثلاً سوف ندخل معلومات لتوافق إما دالة مرفوعة لقوة أو المنحنى الأسى التي تمت مناقشته فيما سبق ، أو كثيرة حدود حتى الدرجة التاسعة (أى لها 10 حدود) - وسوف نسمح بكيفية تصل إلى 100 زوج من البيانات ، وسوف نحول قيم x_i ، y_i إلى $\log x_i$ و $\log y_i$ داخلياً إذا احتجنا لهذه اللوغاريثات . وحيث أن عدد نقط البيانات التي يجب إدخالها ونوع المنحنى المطلوب توفيقه سوف يتنبر من مسألة لأخرى فإننا سوف نستخدم خاصية البعد المتغير عند كتابة البرنامج .

سوف تم طباعة معاملات المعادلات الخطية كجزء من المخرجات . ويمكن أن تكون هذه المعلومات لها أهمية عند تتبع البرنامج لاكتشاف الأخطاء الموجودة به . كذلك فإن معادلة المنحنى المطلوب توفيقه تكون متضمنة أيضاً في المخرجات مع عرض القيم العددية التي تم الحصول عليها لعدد من الكميات الثابتة ، ثم قائمة ببيانات الإدخال x_i و y_i مع القيمة المناظرة لها والتي تم حسابها $y(x_i)$ (وسوف يبسط ذلك عملية رسم بيانات الإدخال والمنحنى الذي تم توفيقه) ، والقيمة العددية لمجموع مربعات الأخطاء . هذه الكمية الأخيرة تفيد في مقارنة نجاح عملية توفيق عدة منحنيات مختلفة لنفس مجموعة البيانات (وكلما كان المجموع قليلاً ، كلما كان التوفيق أصح) .

المخطط التمهيدى للبرامج The Program Outline

من أجل تخطيط البرنامج دعنا نعرف المتغيرات والمجموعات المتراسة التالية :

$$X = \text{متجه له 100 عنصر يحتوى على القيم المدخلة } x_i$$

$$Y = \text{متجه له 100 عنصر يحتوى على القيم المدخلة } y_i$$

$$A = \text{مصفوفة } 10 \times 10 \text{ تحتوى على معاملات الكميات الثابتة المجهولة في المعادلات الخطية .}$$

$$B = \text{مقلوب المصفوفة } A$$

$$C = \text{متجه له 10 عناصر يحتوى على الكميات الثابتة المجهولة في المعادلات الخطية}$$

$$D = \text{متجه له 10 عناصر يحتوى على حدود الطرف الأيمن للمعادلات الخطية .}$$

$$M = \text{الكمية المدخلة التي تشير إلى عدد أزواج البيانات .}$$

$$N = \text{الكمية المدخلة التي تشير إلى المنحنى الواجب استخدامه .}$$

$$N = 0 \text{ يشير إلى الدالة المرفوعة إلى قوة } y = ax^b$$

$$N = 1 \text{ يشير إلى الدالة الأسية } y = ae^{bx}$$

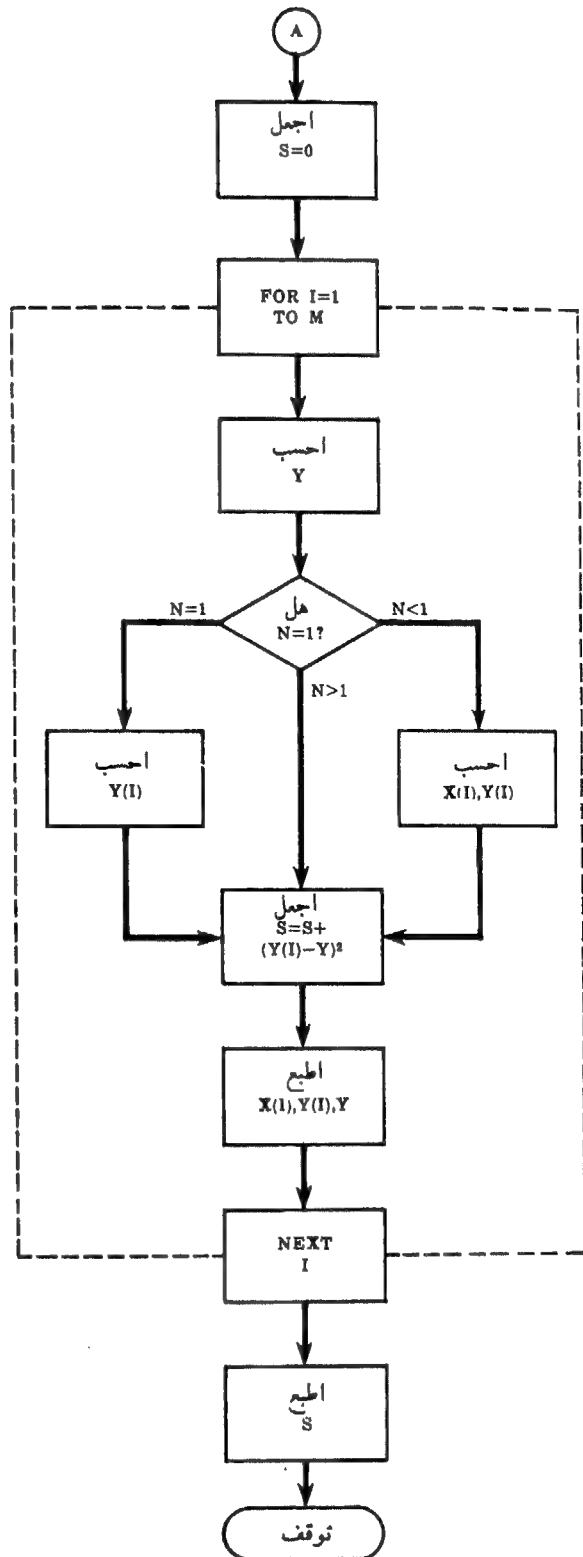
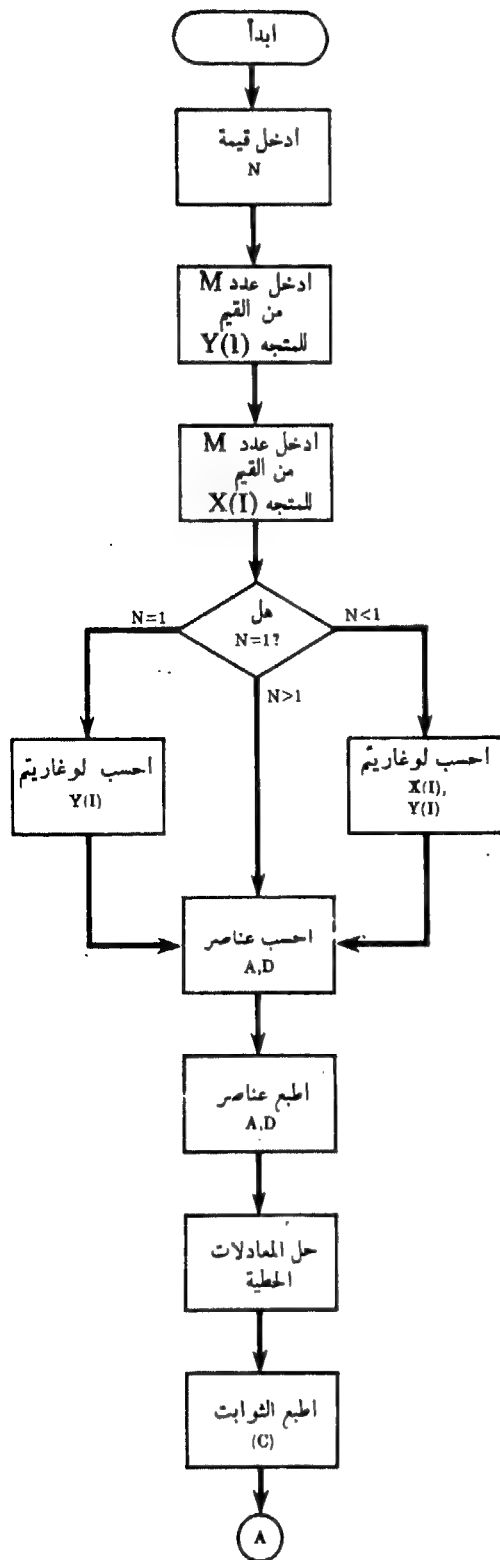
$$N = 2, 3, \dots, 10 \text{ تشير إلى كثيرة الحدود } y = \sum_{i=1}^N c_i x^{i-1}$$

$$(\text{فمثلاً } N = 3 \text{ تشير إلى كثيرة الحدود من الدرجة الثانية : } y = c_1 + c_2 x + c_3 x^2)$$

$$N1 = \text{عدد المعادلات الجبرية الخطية الآتية :}$$

$$N1 = 2 \text{ إذا كانت } N = 0 \text{ أو } N = 1$$

$$N1 = N \text{ إذا كانت } N = 2, 3, \dots, 10$$



شكل ٧ - ٨

نواصل المخطط التمهيدى للبرنامج الأساسى كما يلى :

- ١ - نقرأ بيانات الإدخال .
 - (١) نقرأ قيمة N ، وبذلك نحدد نوعية المنحنى المطلوب توقيقه .
 - (ب) نقرأ عدد M من القيم من أجل i لا يتبها عدد M من القيم من أجل i .
 - ٢ - احسب لوغاريتم x_i ولوغاريتم y_i إذا كانت $N=0$ أو احسب لوغاريتم y_i إذا كانت $N=1$ وذلك لكل قيم $i=1, 2, \dots, M$
 - ٣ - احسب عناصر المصفوفتين A و D ، باستخدام الصيغ الرياضية الملائمة للمنحنى الذى تم اختياره .
 - ٤ - اطبع عناصر المصفوفتين A و D .
 - ٥ - حل المعادلات الجبرية الخطية الآتية .
 - ٦ - اطبع القيم التى تم الحصول عليها للتوابت المجهولة (عناصر المصفوفة C) فى صيغة تبين المعادلة للمنحنى المطلوب .
 - ٧ - احسب $y(x_i)$ لكل قيم $i = 1, 2, \dots, M$
 - ٨ - احسب مجموع مربعات الأخطاء :
- $$e_1^2 + e_2^2 + \dots + e_M^2$$
- ٩ - اطبع قيم x_i و y_i و $y(x_i)$ لكل قيم $i = 1, 2, \dots, M$. ثم بعد ذلك اطبع مجموع مربعات الأخطاء (لاحظ أنه إذا كانت $N=0$ أو $N=1$ فسوف يصبح من المهم تحويل لوغاريتم y_i وربما لوغاريتم x_i مرة أخرى إلى قيم y_i و x_i .
 - ١٠ - توقف .

يبين شكل ٧-٨ خريطة سير العمليات لإجراء الحسابات .

برنامج البيسك The BASIC Program

يبين شكل ٧-٩ برنامج بيسك كامل . هذا البرنامج طويل إلى حد ما ويجب أن يفحص بشئ من العناية . وقد تضمن البرنامج عدداً من جمل REM وذلك من أجل تقسيم البرنامج إلى عدة كتل أساسية منطقية .

لاحظ أن البرنامج قد تضمن عدة جمل مصفوفات وكذلك دالة NUM (انظر السطور 70 و 75 و 90 و 95 و 200 و 205 و 370 و 375).

تسمح جمل MAT INPUT بإدخال عدد متغير من نقط البيانات ، و جمل MAT ZER تسبب إعادة تحديد حجم المصفوفة A والمنتجه D فى كل مرة ينفذ فيها البرنامج . وفى جمل 370 و 375 يعاد تحديد حجم المصفوفة B والمنتجه C ضمناً وذلك حتى تناظر أحجام المصفوفة A والمنتجه D على الترتيب .


```

5  REM  LEAST SQUARES CURVE FITTING
10  DIM X(100),Y(100)
15  PRINT "INPUT N=0 FOR A POWER FUNCTION, N=1 FOR AN EXPONENTIAL."
20  PRINT "FUNCTION."
25  PRINT "FOR A POLYNOMIAL, LET N EQUAL THE NUMBER OF TERMS IN ";
30  PRINT "THE POLYNOMIAL."
35  PRINT "N=";
40  INPUT N
45  REM
50  REM  ENTER DATA POINTS
55  REM
60  PRINT "ENTER THE Y-VALUES"
65  PRINT "ENTER THE Y-VALUES"
70  MAT INPUT Y
75  LET M=NUM
80  PRINT
85  PRINT "ENTER THE X-VALUES"
90  MAT INPUT X
95  IF NUM=0 THEN 115
100 PRINT "THE NUMBER OF Y-VALUES DOES NOT CORRESPOND TO THE ";
105 PRINT "NUMBER OF X-VALUES"
110 STOP
115  REM  CALCULATE LOGARITHMS OF X- AND Y-VALUES IF NECESSARY
120 REM
125 IF N=2 THEN 170
130 FOR I=1 TO M
135 LET Y(I)=LOG(Y(I))
140 NEXT I
145 IF N=1 THEN 170
150 FOR I=1 TO M
155 LET X(I)=LOG(X(I))
160 NEXT I
165 REM
170 REM  CALCULATE ELEMENTS OF A-MATRIX AND D-VECTOR
180 REM
185 LET N1=N
190 IF N1=2 THEN 200
195 LET N1=2
200 MAT A=ZER(N1,N1)
205 MAT D=ZER(N1)
210 FOR I=1 TO N1
215 FOR J=1 TO N1
220 IF I=J THEN 235
225 LET A(I,J)=M
230 GOTO 250
235 LET A(I,J)=M
240 FOR K=1 TO M
245 LET A(I,J)=A(I,J)+X(K)^(I+J-2)
250 NEXT K
255 FOR K=1 TO M
260 IF I=1 THEN 275
265 LET D(I)=D(I)+Y(K)
270 GOTO 280
275 LET D(I)=D(I)+Y(K)*X(K)^(I-1)
280 NEXT K
285 NEXT I
290 REM
295 REM  PRINT SIMULTANEOUS LINEAR EQUATIONS
300

```

```

305 PRINT "COEFFICIENTS IN SYSTEM OF LINEAR EQUATIONS"
310 PRINT
315 PRINT
320 FOR I=1 TO N1
325 FOR J=1 TO N1
330 PRINT A(I,J);
335 NEXT J
340 PRINT D(I);
345 PRINT
350 NEXT I
355 REM  SOLVE SIMULTANEOUS LINEAR EQUATIONS
360 REM
370 MAT B=INV(A)
375 MAT C=B*D
380 REM
385 REM  PRINT EQUATION FOR CURVE FIT
390 IF N=1 THEN 430
395 IF N=2 THEN 430
400 LET C1=EXP(C(1))
405 IF N=1 THEN 420
410 PRINT "POWER FUNCTION: Y=";C1;"*X";C(2)
415 GOTO 490
420 PRINT "EXPONENTIAL FUNCTION: Y=";C1;"*EXP(";C(2);"*X);"
425 GOTO 490
430 IF C(2)>0 THEN 445
435 PRINT "POLYNOMIAL FUNCTION: Y=";C(1);C(2);"X";
440 GOTO 480
445 PRINT "POLYNOMIAL FUNCTION: Y=";C(1);"X";C(2);"X";
450 IF N=2 THEN 485
455 FOR I=3 TO N
460 IF C(I)=0 THEN 475
465 PRINT C(I);"X";I-1;
470 GOTO 480
475 PRINT "X";C(I);"X";I-1;
480 NEXT I
485 PRINT
490 REM  PRINT INPUT VALUES OF X AND Y AND CALCULATED VALUES OF Y
495 REM
500 IF N=2 THEN 545
505 FOR I=1 TO M
510 LET Y(I)=EXP(Y(I))
515 NEXT I
520 IF N=1 THEN 545
525 FOR I=1 TO M
530 LET X(I)=EXP(X(I))
535 NEXT I
540 PRINT
545 PRINT TAB(2);"X";TAB(16);"Y (ACTUAL)";TAB(30);"Y (CALCULATED)"
550 LET S=0
555 FOR I=1 TO M
560 IF N=2 THEN 565
565 IF N=1 THEN 585
570 LET Y1=C1*X(I)^C(2)
575 GOTO 615
580 LET Y1=C1*EXP(C(2)*X(I))
585 GOTO 615
590 LET Y1=C(I)
595 FOR J=2 TO N
600 LET Y1=Y1+C(J)*X(I)^(J-1)
605 NEXT J
610 LET S=S+(Y(I)-Y1)^2
615 PRINT X(I),Y(I),Y1
620 NEXT I
625 PRINT
630 PRINT "SUM OF SQUARE ERRORS=";S
640 END

```

INPUT N=0 FOR A POWER FUNCTION, N=1 FOR AN EXPONENTIAL FUNCTION.
FOR A POLYNOMIAL, LET N EQUAL THE NUMBER OF TERMS IN THE POLYNOMIAL.
N= ?0

ENTER THE Y-VALUES

7.01, .02, .02, .03, .03, .04, .04, .09, .24, .38, .63, .93, 1.24, 1.48, 1.734
72.07, 2.50, 3.12, 3.48

ENTER THE X-VALUES

7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

COEFFICIENTS IN SYSTEM OF LINEAR EQUATIONS

19 51.4244 -26.0334

51.4244 141.871 -56.6213

POWER FUNCTION: $Y = 2.26350E-7 * X^5 5.14716$

X	Y (ACTUAL)	Y (CALCULATED)
7.	0.01	5.06560E-3
8	2.00000E-2	1.00722E-2
9.	2.00000E-2	1.84678E-2
10.	3.00000E-2	3.17640E-2
11.	3.00000E-2	5.18788E-2
12.	4.00000E-2	8.11884E-2
13.	4.00000E-2	0.12258
14.	9.00000E-2	0.179506
15.	0.24	0.256038
16	0.38	0.356922
17.	0.63	0.487632
18.	0.93	0.65443
19.	1.24	0.864418
20.	1.48	1.1254
21.	1.73	1.44693
22.	2.07	1.83839
23.	2.5	2.31103
24.	3.12	2.87701
25.	3.48	3.54972

SUM OF SQUARE ERRORS= 0.614177

شكل ١٠ - (أ)

INPUT N=0 FOR A POWER FUNCTION, N=1 FOR AN EXPONENTIAL FUNCTION.
FOR A POLYNOMIAL, LET N EQUAL THE NUMBER OF TERMS IN THE POLYNOMIAL.
N= ?1

ENTER THE Y-VALUES

7.01, .02, .02, .03, .03, .04, .04, .09, .24, .38, .63, .93, 1.24, 1.48, 1.734
72.07, 2.50, 3.12, 3.48

ENTER THE X-VALUES

7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

COEFFICIENTS IN SYSTEM OF LINEAR EQUATIONS

19 304 -26.0334

304 5434 -213.607

EXPONENTIAL FUNCTION: $Y = 8.53337E-4 * \text{EXP}(0.356011 * X)$

X	Y (ACTUAL)	Y (CALCULATED)
7	0.01	1.03137E-2
8	2.00000E-2	1.47242E-2
9	2.00000E-2	2.10205E-2
10	3.00000E-2	3.00094E-2
11	3.00000E-2	4.28422E-2
12	4.00000E-2	6.11625E-2
13	4.00000E-2	8.73170E-2
14	9.00000E-2	0.124656
15	0.24	0.177942
16	0.38	0.254042
17	0.63	0.362705
18	0.93	0.517806
19	1.24	0.739238
20	1.48	1.05335
21	1.73	1.50664
22	2.07	2.15091
23	2.5	3.07069
24	3.12	4.38379
25	3.48	6.25839

SUM OF SQUARE ERRORS= 10.395

شكل ١٠ - (ب)

۲۳۳

INPUT N=0 FOR A POWER FUNCTION, N=1 FOR AN EXPONENTIAL FUNCTION.
FOR A POLYNOMIAL, LET N EQUAL THE NUMBER OF TERMS IN THE POLYNOMIAL.
N= 75

ENTER THE Y-VALUES
7.01, .02, .02, .03, .03, .04, .04, .09, .24, .38, .63, .93, 1.24, 1.48, 1.734
72.07, 2.50, 3.12, 3.48

ENTER THE X-VALUES
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

COEFFICIENTS IN SYSTEM OF LINEAR EQUATIONS

19 304 5434 18.08

304 5434 105184 394.84

5434 105184 2151370 8773.92

POLYNOMIAL FUNCTION: $Y = 1.73884 - 0.34583 X + 1.65945E-2 * X^2$

X	Y (ACTUAL)	Y (CALCULATED)
7	0.01	0.131157
8	0.02	3.42441E-2
9	0.02	-2.94800E-2
10	0.03	-6.00152E-2
11	0.03	-5.73615E-2
12	0.04	-2.15188E-2
13	0.04	4.75128E-2
14	0.09	0.149733
15	0.24	0.285143
16	0.38	0.453741
17	0.63	0.655528
18	0.93	0.890505
19	1.24	1.15867
20	1.48	1.46002
21	1.73	1.79457
22	2.07	2.1623
23	2.5	2.56322
24	3.12	2.99733
25	3.48	3.46463

SUM OF SQUARE ERRORS= 8.91439E-2

شکل ۷-۱۰ (ج)

INPUT N=0 FOR A POWER FUNCTION, N=1 FOR AN EXPONENTIAL FUNCTION.
FOR A POLYNOMIAL, LET N EQUAL THE NUMBER OF TERMS IN THE POLYNOMIAL.
N= 75

ENTER THE Y-VALUES
7.01, .02, .02, .03, .03, .04, .04, .09, .24, .38, .63, .93, 1.24, 1.48, 1.734
72.07, 2.50, 3.12, 3.48

ENTER THE X-VALUES
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

COEFFICIENTS IN SYSTEM OF LINEAR EQUATIONS

19 304 5434 105184 2151370 18.08

304 5434 105184 2151370 45723424 394.84

5434 105184 2151370 45723424 9.98814E+8 8773.92

105184 2151370 45723424 9.98814E8 2.22672E+10 197719

2151370 45723424 9.98814E+8 2.22672E+10 5.04212E+11 4.50758E+6

POLYNOMIAL FUNCTION: $Y = -0.727783 + 0.330811 * X - 4.78821E-2 * X^2 + 2.56300E-3 * X^3 - 3.63439E-5 * X^4$

X	Y (ACTUAL)	Y (CALCULATED)
7	0.01	3.35158E-2
8	0.02	1.76392E-2
9	0.02	1.03749E-3
10	0.03	-8.82534E-3
11	0.03	-3.35775E-3
12	0.04	2.21596E-2
13	0.04	7.35737E-2
14	0.09	0.15536
15	0.24	0.271119
16	0.38	0.423584
17	0.63	0.614611
18	0.93	0.845186
19	1.24	1.11542
20	1.48	1.42456
21	1.73	1.77098
22	2.07	2.15216
23	2.5	2.56473
24	3.12	3.00446
25	3.48	3.4662

SUM OF SQUARE ERRORS= 6.42682E-2

شکل ۷-۱۰ (د)

تم تبسيط حل المعادلات الآتية باستخدام جمل المصفوفات (انظر الجملتين 370 و 375). ومن ناحية أخرى ، فقد يتطلب البرنامج عديداً من الحلقات التكرارية FOR-TO ، بالرغم من جمل المصفوفات المتاحة . وهناك عدة أسباب لذلك . أولاً ، أن عناصر المصفوفة A تولد داخلياً (السطور من 185 إلى 285) . أيضاً ، يجب تحويل بيانات الإدخال إلى صيغة لوغاريتمية وبالعكس وذلك تحت شروط معينة ، (السطور من 135 إلى 145 ومن 155 إلى 165 ومن 510 إلى 520 ومن 530 إلى 540) . وأخيراً ، صياغة شكل النتائج المطلوب طباعتها تتطلب وجود عدة حلقات تكرارية FOR-TO (السطور من 320-350 ومن 455 إلى 488 ومن 650 إلى 625) وذلك من أجل إخراج البيانات .

تطبيق البرنامج Application of the Program

وبالرجوع إلى المسألة الخاصة بنا وهي سوق الأوراق المالية . دعنا نوفق منحني الأرباح ضد بيانات الزمن للشركة Federated Mousetraps باستخدام دالة مرفوعة لقوة ومنحنى أسى لكثيرة الحدود من الدرجة الثانية ($N=3$) وكثيرة حدود من الدرجة الرابعة ($N=5$) ولكننا نين قبل مواصلة الحسابات أنه في حسابات توفيق منحني يجب أن نراعى عدم تباعد قيم X عن قيم Y . نطرح الرقم 1950 من كل من قيم X وبذلك ، فإن السنة 1957 سوف تمثل بكل بساطة بالرقم 7 مثلاً ، .. وهكذا .

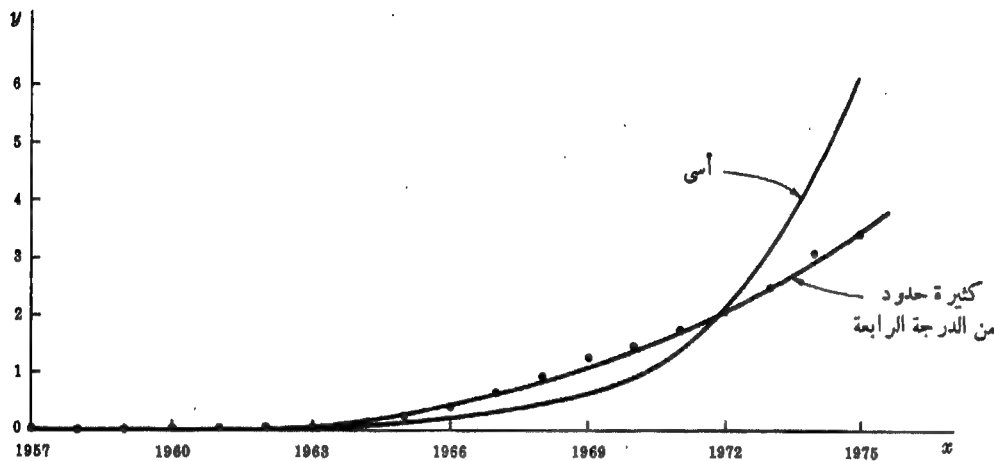
نتائج الحسابات مبينة في الأشكال ٧ - ١٠ (أ) إلى ٧ - ١٠ (د) لكل قيم N تساوى 0 و 1 و 3 و 5 على الترتيب . ويفحص مجموع مربعات الأخطاء في كل حالة ، نرى أن أفضل توفيق هو الذي حصلنا عليها باستخدام كثيرة الحدود من الدرجة الرابعة ($N=5$) منتجة المعادلة :

$$y = -0.727783 + 0.330811x - 0.047882x^2 + 0.002563x^3 - 0.000036x^4$$

(لاحظ أن كثيرة الحدود من الدرجة الثانية أنتجت توفيقاً جيداً تقريباً) . ومن ناحية أخرى كان أسوأ توفيق هو الذي حصلنا عليها باستخدام المنحنى الأسى ($N=1$) . وقد أنتج التعبير الرياضي :

$$y = 0.000853 e^{0.358011x}$$

تم رسم كل من هذه المعادلات بيانياً باستخدام البيانات الأصلية في شكل ٧ - ١١ .



شكل ٧ - ١١

من أجل تقدير الربح لكل سهم لسنة 1978 ، سوف نحسب ($28 = 1978 - 1950$) . بالتعويض بهذه القيمة في كثيرة الحدود من الدرجة الرابعة سوف تنتج القيمة $y = 4.92$. وبذلك نرى أن الأرباح التقديرية لكل سهم من أسهم الشركة Federated Mousetraps Inc سوف تكون \$ 4.29 لسنة 1978 .

وقياساً على متوسط نسبة السعر - للدخل للقيمة 30 (وهي عادية بالنسبة لشركة تنمو بهذه السرعة) ، سوف نقدر أن سهماً من أسهم الشركة (Federated Mousetraps) يجب بيعه تقريباً بمبلغ \$147.60 في سنة 1978 أما إذا كنا قد اشترينا المهزون في 1974 بسعر \$75.00 (مناظر لنسبة السعر للدخل للقيمة 24). فسوف يكون من المعقول أن نتوقع مضاعفة النقود تقريباً في سنة 1978 وذلك يكافئ مكسباً حوالى 19% في السنة بدلالة ربح بسيط يضاعف سنوياً .

كل الأمثلة المعطاة في هذا الفصل كانت تتعلق بالمجموعات المتراسة التي تمثل عناصرها قيماً عديدة . ولكن نذكر القارئ ، أن البيسك يقبل أيضاً المجموعات المتراسة الحرفية وأن جمل المصفوفة يمكن أيضاً استعمالها في القيام بعمليات المجموعات المتراسة الحرفية التقليدية . ومع كل فتذكر أن بعض نسخ البيسك لا تدعم جمل المصفوفة ، وهذا صحيح خصوصاً في نسخ البيسك الجديدة والمتاحة للحاسبات الدقيقة .

اسئلة للمراجعة

Review Questions

- ٧-١ ما هو المنتج ؟ وما هي المصفوفة ؟
- ٧-٢ هل من الضروري استخدام جمل المصفوفات للقيام بعمليات المنتج والمصفوفة ؟ وما هي مزايا استخدام جمل المصفوفات ؟
- ٧-٣ ما هو الفرق بين الضرب غير الموجه وضرب المصفوفة ؟ وما هي الشروط الواجب توافرها من أجل ضرب مصفوفة في أخرى ؟
- ٧-٤ لخص قواعد كتابة ككل من الجمل التالية للمصفوفات :

(أ) تحديد قيم العناصر	(ب) الجمع	(ج) الطرح
(د) الضرب غير الموجه	(هـ) ضرب المصفوفات	
- ٧-٥ ما هو الفرض من جملة MAT READ ؟ وما هي جملة البيسك الأخرى التي يجب أن تستخدم مقترنة بهذه الجملة MAT READ
- ٧-٦ لخص قواعد كتابة جملة MAT READ .
- ٧-٧ نفرض أن جملة MAT READ تحتوى على اسم مصفوفة واحدة . فبأى ترتيب تعطى القيم الموجودة في كتلة البيانات لعناصر المصفوفة ؟
- ٧-٨ كيف يتأثر العنصر رقم صفر المنتج أو المصفوفة باستخدام جملة MAT READ ؟
- ٧-٩ نفرض أن جملة MAT READ واحدة تحتوى على عدة أسماء لمجموعات متراسة . فبأى ترتيب تعطى القيم الموجودة في كتلة البيانات لعناصر المجموعات المتراسة ؟
- ٧-١٠ ما هو الفرض من جملة MAT PRINT ؟ لخص قواعد كتابة هذه الجملة .
- ٧-١١ نفرض أن جملة MAT PRINT تحتوى على اسم مصفوفة واحد . فبأى ترتيب سوف تطبع عناصر المصفوفة ؟ وكيف نميز بين صف في مجموعة وصف آخر ، وكيف يمكن أن نغير في التباعد بين العناصر ؟
- ٧-١٢ نفرض أن جملة MAT READ تحتوى على اسم منتج . فكيف يمكن طباعة عناصر المنتج في شكل عمودي ؟ وفي شكل صف ؟ وكيف يمكن تغيير التباعد بين عناصر المنتج ؟
- ٧-١٣ كيف تتأثر العناصر التي تحمل أرقام صفر في منتج أو مصفوفة بجملة MAT PRINT ؟
- ٧-١٤ نفرض أن جملة MAT PRINT واحدة تحتوى على عدة أسماء لمجموعات متراسة . فبأى ترتيب سوف تتم طباعة عناصر المجموعة المتراسة ؟ وكيف يمكن التمييز بين عناصر مجموعة متراسة وعناصر مجموعة متراسة أخرى ؟
- ٧-١٥ هل يمكن تضمين الصيغ الرياضية أو مراجع الدوال في جملة MAT PRINT ؟
- ٧-١٦ ما هو الفرض من جملة MAT INPUT ؟ وكيف تختلف عن جملة MAT READ ؟

- ٧-١٧ ملخص قواعد كتابة جملة MAT INPUT .
- ٧-١٨ ماذا يحدث عند تنفيذ جملة MAT INPUT ؟ وكيف تنقل قيم البيانات المدخلة للحاسب ؟
- ٧-١٩ كيف يتأثر العنصر رقم صفر بواسطة جملة MAT INPUT ؟
- ٧-٢٠ كيف يمكن قراءة سطر أو أكثر من بيانات الإدخال بواسطة جملة MAT INPUT واحدة ؟
- ٧-٢١ كيف يمكن تحديد عدد عناصر المتجه التي تم إدخالها بواسطة جملة MAT INPUT اقترح مكاناً ملائماً لتخزين هذا الرقم .
- ٧-٢٢ هل يمكن لجملة MAT INPUT أن تستخدم لإدخال عناصر مصفوفة كما هو الحال مع عناصر المتجه ؟ هل هناك أى قيود على المصفوفات التي لا توجد مع المتجهات ؟
- ٧-٢٣ ما هي مصفوفة الوحدة ؟
- ٧-٢٤ ما هو المقصود من تدوير مصفوفة ؟
- ٧-٢٥ ما هو المقصود من إيجاد مقلوب المصفوفة ؟ هل يمكن الحصول على مقلوب المصفوفة دائماً ؟ اشرح .
- ٧-٢٦ ما هي العلاقة التي يجب أن توجد ما بين أرقام الصفوف وأرقام الأعمدة في مصفوفة الوحدة ؟ وفي مقلوب مصفوفة ؟ (أو مصفوفة مطلوب إيجاد مقلوبها) ؟
- ٧-٢٧ ما هي العلاقة التي يجب أن توجد ما بين أرقام السطور وأرقام الأعمدة لمصفوفة معينة والمصفوفة الناتجة من تدويرها ؟
- ٧-٢٨ ما هي نتيجة ضرب مصفوفة بمصفوفة الوحدة ؟
- ٧-٢٩ ما هي نتيجة ضرب مصفوفة بمقلوبها ؟
- ٧-٣٠ اذكر الغرض وملخص قواعد كتابة كل من جمل المصفوفات التالية :
- (a) MAT ZER
(b) MAT CON
(c) MAT IDN
(d) MAT TRN
(e) MAT INV
- ٧-٣١ ما هو الغرض من الدالة المكتبية DET ؟ ولأى نوع من أنواع المصفوفات يمكن أن تستخدم دالة DET ؟ وما هي جملة المصفوفة التي يجب أن تسبق دائماً الإشارة إلى دالة DET ؟
- ٧-٣٢ كيف يمكننا أن نقرر ما إذا كانت المصفوفة المربعة يمكن الحصول على مقلوبها أم لا ؟
- ٧-٣٣ كيف يمكننا حل المعادلات الجبرية الخطية الآتية باستخدام جمل المصفوفة ؟ هل يوجد إجراء معين للتحقق من دقة الحل ؟
- ٧-٣٤ ما هي جمل المصفوفة التي تسمح بإعادة تحديد حجم مجموعة المتراصة أثناء تنفيذ البرنامج ؟ وكيف يمكن إنجاز إعادة تحديد حجم المجموعة المتراصة ؟
- ٧-٣٥ قارن بين مفهوم إعادة تحديد حجم المجموعة المتراصة التي تمت مناقشتها في القمم ٧-٤ وبين خاصية البعد المتغير المتضمن في جملة MAT INPUT .

٣٦-٧ ما هي العمليات التي يمكن تأديتها على المجموعات المتراسة الحرفية ؟ وما هي جمل المصفوفة التي يمكن استخدامها في مناولة المجموعات المتراسة الحرفية ؟

مسائل محلولة Solved Problems

٣٧-٧ مبين فيما يلي عدة جمل أو مجموعات من جمل البيسك . بعض منها مكتوب بصورة غير صحيحة تعرف على كل الأخطاء :

10 DIM P(12,20),Q(12,20),R(12,20) (أ)

50 MAT P=(.5)*(Q+R)

لا يمكن أن تظهر صيغ رياضية للمصفوفة في جملة المصفوفة .

10 DIM P(12,20),Q(12,20) (ب)

50 MAT P=((N+1)/2)*Q

صحيح ، مادامت N متغير عادي (غير متجه) .

30 MAT INPUT A,B,C (ج)

معظم نسخ البيسك تسمح بظهور اسم متجه واحد فقط في جملة MAT INPUT .

30 MAT INPUT N\$ (د)

صحيحة ، مادامت N\$ هي متجه حرفي .

200 MAT PRINT A,B,A+B,A-B (هـ)

لا يمكن أن تظهر الصيغ الرياضية في جملة MAT PRINT

10 DIM A(10,20),B(20,10) (و)

30 MAT READ A

40 MAT B=TRN(A)

صحيحة

10 DIM P(3,6),Q(8,10),R(5,5) (ز)

50 MAT R=P+Q

لا يمكن جمع المصفوفات مادامت أبعادها غير متناظرة .

10 DIM C(6,8),D(6,8),V(12) (ح)

60 MAT D=INV(C)

90 MAT V=IDN

من الاستحالة حساب مقلوب مصفوفة غير مربعة . وأيضاً لا يمكن أن يظهر متجه في جملة MAT IDN

٢٨-٧ اكتب جملة أو أكثر من جمل البيسك لكل من المواقف الموصوفة التالية :

(أ) احسب قيمة الصيغة الرياضية

$$Y = X^T * A * X$$

حيث A مصفوفة (10×10) و X : من (10) عناصر (وذلك بإهمال المنصر رقم صفر) و X^T هي مصفوفة X المرددة . ما هي أبعاد Y ؟

```
100 MAT Z=TRN(X)
110 MAT B=Z*A
120 MAT Y=B*X
```

أو

```
100 MAT Z=TRN(X)
110 MAT C=A*X
120 MAT Y=Z*C
```

(سوف تمثل Y قيمة واحدة (أى أن Y سوف تكون متغيراً عادياً وليس مجموعة مترابطة)

(ب) احسب قيمة الصيغة الرياضية :

$$T = (N) * F^{-1} * G + H$$

حيث F و G و H مصفوفات (50×50) و N متغير عادى (غير متجه) و F^{-1} تمثل مقلوب المصفوفة F . ما هي أبعاد T ؟

```
100 MAT A=INV(F)
110 MAT B=A*G
120 MAT C=(N)*B
130 MAT T=C+H
```

سوف تكون T مصفوفة (50×50)

(ج) احسب الفرق بين I و $A^{-1} * A$ حيث A مصفوفة (10×10) و A^{-1} مصفوفة تمثل مقلوب المصفوفة A و I هي مصفوفة الوحدة (10×10) (لاحظ أن هذا الفرق من حيث المبدأ يجب أن يكون مصفوفة عناصرها صفراً . وفي الحقيقة فإن عناصر المصفوفة يمكن أن تختلف عن الصفر وذلك بسبب الأخطاء الرقمية) .

```
100 MAT B=INV(A)
110 MAT C=B*A
120 MAT I=IDN
130 MAT D=I-C
```

(د) اطبع الإجابة التي تم الحصول عليها في الجزء (ج) السابق في شكل مصفوفة ، مع تقارب عناصرها على قدر المستطاع .

```
140 MAT PRINT D;
```

(هـ) نفرض أن A و B مصفوفتان (10×10) حيث تعطى عناصرها في كتلة بيانات . اقرأ عناصر A ، صفناً في كل مرة ، ثم تتيبها عناصر B عموداً في كل مرة .

```
10 MAT READ A,C
20 MAT B=TRN(C)
```

(لاحظ أن A و C نقرأ صف صف)

(و) نفرض أن X و Y متجهات يجب أن تحدد قيم عناصرها بقيم تغطي من النهاية الطرفية المركزية . وسوف نخزن عدد القيم التي تم إدخالها لكل من X و Y في $X(0)$ و $Y(0)$ على الترتيب .

```
10 MAT INPUT X
20 LET X(0)=NUM
30 MAT INPUT Y
40 LET Y(0)=NUM
```

(ز) نفرض أن P مصفوفة (5×12) و T مصفوفة (6×6) و Y متجه له (10) عناصر . اطبع قيم P و T و X حيث تظهر قيم X في شكل صف .

```
200 MAT PRINT P,T,X,
```

(ح) كرر الجزء (ز) السابق ، مع تقارب عناصر المجموعة المتراصة على قدر المستطاع .

```
200 MAT PRINT P,T,X;
```

مسائل تكميلية

Supplementary Problems

٧ - ٣٩ مبين فيما يلي عدة جمل أو مجموعات من جمل البيسك . بعضها كتب بطريقة خاطئة . تعرف على كل الأخطاء :

```
10 DIM L$(100),M$(100) ( ا )
```

```
...
50 MAT INPUT M$
60 MAT L$=M$
```

```
35 MAT Q=(A+B)*P ( ب )
```

حيث A و B و P و Q مصفوفات (10×10)

```
35 MAT Q=(A+B)*P ( ج )
```

حيث A و B متغيرات عادية (غير متجه) و P و Q مصفوفات (10×10)

```
60 MAT H=(3)*H ( د )
```

حيث H مصفوفة 6×6

```
60 MAT G=H*H ( هـ )
```

حيث G و H مصفوفتان (6×6)

```
60 MAT H=G*H ( و )
```

حيث G و H مصفوفتان (6×6)

```
10 DIM X(10,20),Y(10,20),Z(10,20) ( ز )
```

```
...
50 MAT Z=X*Y
```

```
10 DIM V(100) ( ح )
```

```
...
50 MAT INPUT V(50)
```

10 DIM C(100,50),D(100,50) (ط)
20 INPUT M,N
30 MAT READ C(M,N),D(M,N)

10 DIM C(100,50),D(100,50) (ى)
20 INPUT M,N
...
100 MAT PRINT C(M,N),D(M,N)

10 DIM A(25,25) (ك)
...
50 MAT A=CON(12,24)

10 DIM A(20,5) (ل)
...
80 MAT A=ZER(10,6)

10 DIM F(10,20),G(10,20) (م)
...
60 MAT G=INV(F)

10 DIM K(20,30) (ن)
...
50 MAT K=IDN(12,12)

٧-٤ : اكتب جملة أو أكثر من جمل البيسك لكل من المواقف الموصوفة فيما يلي :

(١) احسب قيمة الصيغة الرياضية .

$$F=(2*N+1)*(A^T*A-I)$$

حيث A مصفوفة (10×10) و A^T مصفوفة A المدورة ، I هي مصفوفة الوحدة (10×10) و N متغير عادي (غير متجه) . ماذا تكون أبعاد F ؟

(ب) افرض أن A و B و C و D مصفوفات 10×10 احسب محدد G حيث :

$$G=A*C-B*D$$

ثم اطبع نتيجته .

(ج) عدل جزء (ب) السابق وذلك لطباعة عناصر G يتبعها مقلوب G ثم محدد G . قارب بين عناصر المصفوفة على قدر المستطاع .

(د) نفرض أن A مصفوفة (20×30) حدد قيم أول 8 أعمدة من أول 12 صف بصفر .

(هـ) نفرض أن A و B مصفوفتان (20×30) . اقرأ العناصر في الأعمدة الإثني عشر الأولى من الثمانية صفوف الأولى في المصفوفة A ، تتبعها عناصر الخمسة عشرة الأولى من الستة صفوف الأولى من المصفوفة B . كيف يجب تنظيم البيانات في كتلة البيانات ؟

(و) نفرض أن A و B مصفوفتان (20×30) اطبع العناصر الموجودة في الإثني عشر عموداً الأولى من الثمانية صفوف الأولى من المصفوفة A ، تتبعها عناصر الأعمدة الخمسة عشر الأولى من الستة صفوف الأولى من المصفوفة ثم قارن بالحل الموجود في الجزء (هـ) السابق .

مسائل للبرمجة Programming Problems

٧-١١ عدل البرنامج المعطى في مثال ٧-١٥ حيث يمكن حساب المصفوفة المدورة للمصفوفة F، ثم مقلوبها ثم محدد المصفوفة F واطبمها. اشرح النتائج التي سوف تحصل عليها.

٧-٢٢ عدل البرنامج المعطى في مثال ٧-١٩ حتى يمكن حل أى عدد من N معادلة في N مجهول. استخدم خاصية البعد المتغير الموصوف في قسم ٧-٤. أدخل وسيلة لطباعة محدد معاملات المصفوفة.

٧-٢٣ عند توليد أخطاء رقمية في حل المعادلات الآتية $C \cdot X = D$ فن اليسر أن توصل كما يلي :

$$(١) \quad F = C \cdot Y$$

(ب) احسب متجه الأخطاء $G = D - F$ (إذا لم توجد أخطاء رقمية فإن F ستبقى كما هي كالمتجه المعطى D ، وعناصر G سوف تكون أصفاراً).

(ج) إن لم تكن عناصر G كلها أصفار فعمل المعادلات الآتية التالية :

$$A \cdot Y = G$$

للقم $Y(1)$ و $Y(2)$ و و $Y(N)$

(د) اجمع عناصر Y على العناصر المناظرة في X وذلك للحصول على حل معدل.

عدل البرنامج المعطى في مثال ٧-١٩ متضمناً خاصية تصحيح الأخطاء. أدخل وسيلة لحساب محدد معاملات المصفوفة.

$$\begin{aligned} x_1 + \frac{1}{2}x_2 + \frac{1}{8}x_3 + \frac{1}{4}x_4 + \frac{1}{8}x_5 &= \frac{1}{8} \\ \frac{1}{2}x_1 + \frac{1}{8}x_2 + \frac{1}{4}x_3 + \frac{1}{8}x_4 + \frac{1}{8}x_5 &= \frac{1}{8} \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{8}x_3 + \frac{1}{8}x_4 + \frac{1}{4}x_5 &= \frac{1}{8} \\ \frac{1}{4}x_1 + \frac{1}{8}x_2 + \frac{1}{8}x_3 + \frac{1}{4}x_4 + \frac{1}{8}x_5 &= \frac{1}{8} \\ \frac{1}{8}x_1 + \frac{1}{8}x_2 + \frac{1}{4}x_3 + \frac{1}{8}x_4 + \frac{1}{8}x_5 &= \frac{1}{16} \end{aligned}$$

استخدم البرنامج لحل المعادلات الآتية :

قارن الإجابات التي تم الحصول عليها بدون خاصية تصحيح الأخطاء.

٧-٢٤ عدل البرنامج المعطى في مثال ٧-٢٢ حتى يمكن توفير عدة منحنيات مختلفة لنفس مجموعة البيانات (أى يمكن أن يشغل البرنامج عدة مرات مختلفة) بدون إعادة قراءة البيانات كل مرة يعاد فيها البرنامج.

٧-٢٥ اكتب برنامج يبسك لكل مسألة موصوفة فيما يلي. ضمن الحل مخطط تفصيل أو خريطة لسير عمليات للمسألة.

(١) مبتدئاً بالمصفوفة A الممطاة في مثال ٧-١٨ ، احسب مصفوفة D جديدة حيث كل عنصر من عناصر المصفوفة D هو مقلوب العنصر المناظر في المصفوفة A أى $d_{ij} = 1/a_{ij}$. قارن عناصر المصفوفة D بعناصر مقلوب المصفوفة A (مقلوب المصفوفة ممثل في المصفوفة B في المثال ٧-١٨).

(ب) احسب مقلوب المصفوفة B الممطاة في مثال ٧-١٨. قارن عناصر هذه المصفوفة بعناصر المصفوفة A في المثال

٧-١٨.

(ج) وضح أن $A \cdot B$ لا تساوى $B \cdot A$ لكل من المصفوفتين A و B الممطاة في المثال ٧-١٥.

(د) وضع أن $(A*B)^T$ تساوى B^T*A^T لكل من المصفوفتين A و B المعطيتين في مثال ٧ - ٦ ، حيث A^T تمثل المصفوفة A المدورة . و B^T تمثل المصفوفة B المدورة .. إلخ .

(هـ) احسب قيمة عناصر المصفوفة :

$$Y=D^T*CT*C*D$$

باستخدام المصفوفة C والمتجه D المعطاة في مثال ٧ - ١٩ . ما هي أبعاد Y ؟

(و) احسب قيمة عناصر المصفوفة

$$P=I+A+A^2+A^3+A^4$$

باستخدام المصفوفة A في مثال ٧ - ١٨ (لاحظ أن I تمثل مصفوفة الوحدة و $A^2 = A*A$ و $A^3 = A*A*A$... إلخ) ما هي أبعاد P ؟

الفصل ٨

ملفات البيانات

Data Files

الملف في اللغة الاصطلاحية عبارة عن مجموعة منظمة ومستقلة من المعلومات . ويمكن أن يتضمن أى نوع من المعلومات . من ثم فيمكن أن يحتوى الملف على سلسلة متعاقبة من جمل بيسك ، أو يتكون من قيم البيانات (أى أرقام وحروف) . النوع الأول هو بالطبع برنامج بيسك ويسمى الأخير ملف بيانات .

تعتبر ملفات البيانات وسيلة مناسبة لتخزين مجموعة من البيانات حيث يمكن قراءتها وتعديلها بواسطة برنامج بيسك ويتناول هذا الفصل إنشاء واستخدام مثل هذه الملفات .

وسوف نرى أن البيسك يتضمن عدداً من الجمل للتعامل مع الملفات ، ماثلة لجمل المصفوفات التى تم تقديمها في الفصل السابع . إلا أنها تختلف عن جمل المصفوفات في وجود كثير من الاختلافات في الشكل من نسخة بيسك لأخرى . ولذلك فإن هذا الفصل سوف يؤكد مفهوم التعامل مع ملف البيانات دون الدخول في تفاصيل الجمل كل على حدة . (ظهور جمل التعامل مع الملف في هذا الفصل جزء من لغة بيسك المطبقة على حاسب Digital Equipment Corporations أى نظام DEC-10 . وهى نموذج لجمل التعامل مع الملفات المتاحة في نسخ أخرى من البيسك . أما إجراءات التعامل مع الملفات المستخدمة مع الحاسبات الدقيقة فيتم تقديمها في الفصل التاسع بالقسم ٩ - ٤ .

٨ - ١ ملفات البيانات التسلسلية SEQUENTIAL DATA FILES

من أهم خصائص الملفات التسلسلية في واقع الأمر أن عناصر البيانات تنظم كلا على حدة بطريقة تسلسلية ، أى واحدة تلو الأخرى . وعادة تناظر عناصر هذا النوع من الملفات السطور المنفصلة من المعلومات على وحدة الكونحول . وبذلك فإن برنامج بيسك سوف يخزن كلف تسلسل ، حيث كل جملة سوف تظهر على سطر منفصل وتنظم الجمل بترتيب تصاعدي لأرقام السطور .

يمكن أن تمثل الملفات التسلسلية أيضاً مجموعات من البيانات . وسوف تتكون مثل هذه الملفات من عدة سطور من البيانات ، كل سطر منها يبدأ برقم سطر . وسوف تنظم السطور في تسلسل بترتيب تصاعدي لأرقام السطور . يمكن أن تكون عناصر البيانات في سطر أرقاماً أو حروفاً أو توافقية من الاثنين ، يفصل بين كل منهما فاصلة (,) . أو أماكن خالية إذا كانت سلسلة الحروف بها فاصلة (.) أو مكان خال ، فيجب أن تحصر بين علامتى اقتباس .

مثال ٨ - ١

يحتوى ملف بيانات تسلسل على الاسم ودرجات الاختبارات لكل طالب في فصل دراسي يدرس علوم الحاسب . سوف يظهر الملف كما يلي :

10 "COMP SCI 100" "FALL, 1982"
 20 "ADAMS B F" 45 80 80 95 55
 30 "BROWN P" 60 50 70 75 55
 40 "DAVIS R A" 40 30 10 45 60
 50 "FISHER E K" 0 5 5 0 10
 60 "HAMILTON S P" 90 85 100 95 90
 70 "JONES J J" 95 90 80 95 85
 80 "LUDWIG C W" 35 50 55 65 45
 90 "OSBORNE T" 75 60 75 60 70
 100 "PRINCE W F" 85 75 60 85 90
 110 "RICHARDS E N" 50 60 50 35 65
 120 "SMITH M C" 70 60 75 70 55
 130 "THOMAS B A" 10 25 35 20 30
 140 "WOLFE H" 25 40 65 75 85
 150 "ZORBA D R" 65 80 70 100 60

نرى أن كل سطر يبدأ برقم للسطر . يحتوى السطر الأول على سلسلتين من الحروف ، لتعرف المادة الدراسية والفصل الدراسي على الترتيب . (لاحظ أن سلسلة الحروف تحصر بين علامتي الاقتباس ، حيث تحتوى كل من السلسلتين على الفاصلة (و) والمسافة الخالية) . يحتوى كل من السطور المتعاقبة على اسم الطالب (سلسلة من الحروف) ، تتبعها خمس درجات لامتحانات . وبذلك فإن الملف يتكون من 15 سطراً ، بالرغم من وجود 14 طالباً فقط في الفصل .

ويجب أن يكون مفهوماً أن تتابع ملف البيانات الذي نحن بصدده يتقرر بأرقام السطور وليس بالأسماء ، ولا أهمية في الحقيقة أن تكون الأسماء مرتبة ترتيباً هجائياً .

انشاء وتنقيح ملف متتابع Creating and Editing a Sequential File

لما كان ملف البيانات يبنى بنفس طريقة بناء برنامج بيسك فيمكننا إنشاء وتنقيح مثل هذا الملف بنفس الطريقة المتبعة من البرنامج . ويمكننا أيضاً طباعة ملف بيانات متتابع على الكونسول إذا رغبت في ذلك . تنفذ هذه الوظائف مع أوامر النظام المبثلة في الفصل الثالث (مثال ذلك NEW و OLD و SAVE و LIST و وإلخ) .

قواعد إعطاء اسم لملف البيانات هي نفس القواعد المستخدمة عند إعطاء اسم لبرنامج بيسك . والصورة التقليدية لذلك هي من حرف إلى ستة أحرف ويجب أن تبدأ بحرف هجائي

مثال ٨ - ٢

نفرض أننا نريد إدخال ملف البيانات المتتابع المين في مثال ٨ - ١ للحاسب من خلال الكونسول وتخزينه تحت اسم SCORES ولتنفيذ ذلك يجب أن نسجل دخولنا للنظام ، ثم نعرف أن اسم الملف الجديد هو SCORES ، ثم نطبع مجموعة البيانات سطراً بسطر ثم بعد ذلك نخزن الملف . ويمكننا أيضاً ، إذا رغبت في ذلك ، عمل قائمة بالملف ، بعد إتمام إدخاله وتصحيحه .

يبين شكل ٨ - ١ الجزء الأساسي من جلسة المشاركة الزمنية التي تبدأ بعد إجراء التسجيل لدخول النظام . لاحظ حدوث ثلاثة أخطاء طباعة عند إدخال البيانات (في السطور 120 و 130 و 150) تم تصحيح هذه الأخطاء إما من خلال استخدام مفتاح DELETE (السطور 120 و 130) أو بإعادة طبع السطر بالكامل (انظر السطر 150) .

قائمة كاملة بملف البيانات مبينة في نهاية الشكل ٨ - ١ . تنتج هذه القائمة بناء على أمر LIST ، المين في منتصف الشكل . وكل أوامر النظام التي يدخلها المستخدم موضوع تحتها خط .

يمكن أيضاً إنشاء ملف البيانات المتتابع مباشرة بواسطة برنامج البيسك وسوف نرى كيف يتم تنفيذ ذلك في مثال ٨ - ٤ .

قراءة ملف بيانات متتابع Reading a Sequential Data File

في معظم التطبيقات سوف تقرأ المعلومات المخزنة في ملف البيانات ثم تشغل بواسطة برنامج يبسك . ويجب أن تقرأ بنود ، وتبدأ من ملف البيانات المتتابع بنفس طريقة التخزين وببنفس الترتيب . مبتدئاً من بداية ملف البيانات . وسوف تحفظ البيانات التي تمت قراءتها لاستخدامها بعد ذلك .

ومن أجل قراءة البيانات من ملف بيانات متتابع وتحت تحكم البرنامج ، سوف نستخدم جمل التعامل مع الملفات وهي FILES و INPUT و IF END ويوضح المثال التالي كيف يتم إنجاز ذلك .

```

NEW OR OLD-->NEW
NEW FILE NAME-->SCORES
>10 "COMP SCI 100" "FALL, 1982"
>20 "ADAMS B F" 45 80 80 95 55
>30 "BROWN P" 60 50 70 75 55
>40 "DAVIS R A" 40 30 10 45 60
>50 "FISHER E K" 0 5 5 0 10
>60 "HAMILTON S P" 90 85 100 95 90
>70 "JONES J J" 95 90 80 95 85
>80 "LUDWIG C W" 35 50 55 65 45
>90 "OSBORNE T" 75 60 75 60 70
>100 "PRINCE W F" 85 75 60 85 90
>110 "RICHARDS E N" 50 60 50 35 65
>120 "SMITH M C" 70 60 75 70 55
>130 "THOMAS B A" 10 25 35 20 30
>140 "WOLFE H" 25 40 65 75 85
>150 "ZORBA D R" 65 80 70 100 60
>SAVE
>LIST
SCORES
10 "COMP SCI 100" "FALL, 1982"
20 "ADAMS B F" 45 80 80 95 55
30 "BROWN P" 60 50 70 75 55
40 "DAVIS R A" 40 30 10 45 60
50 "FISHER E K" 0 5 5 0 10
60 "HAMILTON S P" 90 85 100 95 90
70 "JONES J J" 95 90 80 95 85
80 "LUDWIG C W" 35 50 55 65 45
90 "OSBORNE T" 75 60 75 60 70
100 "PRINCE W F" 85 75 60 85 90
110 "RICHARDS E N" 50 60 50 35 65
120 "SMITH M C" 70 60 75 70 55
130 "THOMAS B A" 10 25 35 20 30
140 "WOLFE H" 25 40 65 75 85
150 "ZORBA D R" 65 80 70 100 60

```

شكل ٨ - ١

مثال ٨ - ٣

دعنا نرجع إلى قائمة الأسماء ودرجات الامتحان المبينة في شكل ٨ - ١ . ونفرض أننا نريد القيام بالمعاملات التالية لكل طالب في الفصل .

١ - اقرأ درجات الاختبارات الخمسة الأوائل من ملف بيانات SCORES

٢ - ثم أدخل درجة اختبار سادسة من الكونسول .

٣ - احسب المتوسط لكل من الدرجات الست المعطاة .

٤ - اطبع المتوسط على الكونسول .

وسوف تتوقف الحسابات عند الوصول إلى نهاية ملف البيانات .

وسوف يقوم برنامج بيسك بالعمليات المبينة في شكل ٨ - ٢ . وتعرف المتغيرات الموجودة في البرنامج كالتالي :

N = رقم السطر لكل سطر من ملف البيانات SCORES

T\$ = عنوان المادة الدراسية :

Y\$ = الفصل الدراسي .

N\$ = اسم الطالب

C1 و C2 و C3 و C4 و C5 = درجات الاختبارات لكل طالب في ملف البيانات SCORES

C6 = درجة الاختبار التي تدخل من الكونسول لكل طالب .

A = المتوسط المحسوب للاختبارات الستة للطالب .

فيما يلي شرح لجمل التعامل مع الملف التي تظهر في هذا البرنامج .

```

10 FILES SCORES
20 INPUT #1,N,T$,Y$
30 PRINT "COURSE:";T$,"TERM:";Y$
40 PRINT
50 INPUT #1,N,N$,C1,C2,C3,C4,C5
60 PRINT N$,"SCORE=";
70 INPUT C6
80 LET A=(C1+C2+C3+C4+C5+C6)/6
90 PRINT "AVERAGE=";A
100 PRINT
110 IF END #1,THEN 130
120 GOTO 50
130 END

```

شكل ٨ - ٢

يتم تبادل المعلومات بين ملف البيانات وبرنامج البيسك من خلال قناة البيانات . تحدد الجملة رقم 10 (FILES) ملف البيانات SCORES لقناة البيانات رقم 1 . ويجب أن يحدث ذلك قبل تبادل المعلومات من أو إلى ملف البيانات . وكل جمل التعامل مع البيانات التي تعقب ذلك سوف تشير إلى ملف البيانات برقم القناة وليس باسمه .

تقرأ الجملة رقم 20 رقم السطر N وعنوان المادة الدراسية T\$ والفصل الدراسي Y\$ من أول سطر من ملف البيانات الذي تحددت له قناة رقم 1 ، (أى ملف البيانات SCORES) . تقرأ الجملة 50 ، رقم سطر واسم الطالب و 5 درجات اختبار من كل سطر في SCORES (لاحظ أن أرقام السطور يجب أن تقرأ من ملف البيانات ، بالرغم من أنها لا تستخدم) .

تختبر الجملة 110 نهاية ملف البيانات . إذا لم توجد أى بيانات أخرى (نهاية الملف) ، فإن التحكم سوف يتحول إلى الجملة 130 (END) وإلا فسوف يتحول التحكم إلى الجملة المنفذة التالية (الجملة 120) . وبذلك نرى أن هذه الجملة تشبه كثيراً جملة IF-THEN الشهيرة ، بالرغم من أنها تستخدم فقط لتختبر شرط نهاية الملف .

باقى جمل هذا البرنامج عبارة عن جمل بيسك عادية ، تماماً كالتى نوقشت في فصول سابقة من هذا الكتاب ، ومعانيها يجب أن تكون واضحة .

يبين شكل ٨ - ٣ المخرج الناتج من هذا البرنامج وملف البيانات SCORES . (قائمة بملف البيانات مبين في شكل ٨ - ١) . لاحظ أن المعلومات التي تم إدخالها مباشرة من النهاية الطرفية المركزية هي درجات مادة واحدة لكل طالب . وباقى البيانات كانت محولة للنهية الطرفية المركزية من ملف البيانات أو مولدة بواسطة البرنامج (البيانات المدخلة موضوع تحتها خط) .


```

COURSE: COMP SCI 100          TERM: FALL, 1982
ADAMS B F      SCORE= 723
AVERAGE= 71.6667
BROWN P        SCORE= 780
AVERAGE= 45
DAVIS R A      SCORE= 733
AVERAGE= 40
FISHER E K     SCORE= 71
AVERAGE= 4.16667
HAMILTON S P   SCORE= 790
AVERAGE= 91.6667
JONES J J      SCORE= 780
AVERAGE= 87.5
LUDWIG C W     SCORE= 720
AVERAGE= 53.3333
OSBORNE T      SCORE= 780
AVERAGE= 70
PRINCE W F     SCORE= 7100
AVERAGE= 82.5
RICHARDS E N   SCORE= 770
AVERAGE= 55
SMITH M C      SCORE= 775
AVERAGE= 47.5
THOMAS B A     SCORE= 710
AVERAGE= 21.6667
WOLFE H        SCORE= 793
AVERAGE= 64.1667
ZORBA D R      SCORE= 793
AVERAGE= 78.3333

```

شكل ٨ - ٣

كتابة ملف بيانات متتابع Writing a Sequential Data File

يمكن كتابة المعلومات على ملف البيانات بواسطة برنامج بيسك ، بطريقة تشبه قراءة البيانات من ملف البيانات . عند الكتابة لملف بيانات متتابع ، توضع المعلومات الجديدة أتوماتيكياً وراء البيانات الموجودة ، وبذلك نحصى البيانات الموجودة فعلاً في الملف . إذا كان من الواجب إلغاء البيانات القديمة قبل كتابة بيانات جديدة ، فيجب أن يلغى الملف بصراحة ويماد وضمه إلى نقطة بدايته .

سوف نرى في المثال التالي كيف يمكن أن تقرأ البيانات من ملف بيانات متتابع ومن الكونسول ، ثم تشغل وتكتب النتائج بعد ذلك على الكونسول وعلى ملف بيانات متتابع جديد ، ولإنجاز ذلك سوف نستفيد بحمل التعامل مع الملف وهي FILES و QUOTE و SCRATCH و INPUT و PRINT و IF END . وسوف تتم مناقشة الفرض من كل جملة في المثال .

مثال ٨ - ٤ تشغيل درجات اختبارات طالب Processing Student Examination Scores

سوف نستكمل في هذا المثال المسألة الموصوفة في الأمثلة ٨ - ٢ و ٨ - ٣ ، والتي تناول تسجيل وتشغيل مجموعة من درجات اختبار فصل من الطلبة . دعنا الآن نطور البرنامج حتى يمكنه بصورة أتم تأدية العمليات التالية لكل طالب :

١ - يقرأ مجموعة من درجات اختبار من ملف بيانات موجود (مثال SCORES) .

٢ - يدخل درجة اختبار جديدة من النهاية الطرفية المركزية .

- ٣ - بحسب متوسطاً لكل درجات الاختبارات .
- ٤ - يطبع المتوسط المحسوب على النهاية الطرفية المركزية .
- ٥ - يكتب مجموعة جديدة من البيانات (أى درجات الاختبارات الأساسية ودرجة الاختبار الجديدة والمتوسط المحسوب) على ملف بيانات جديد .
- عند استكمال الحسابات فإن ملف البيانات الجديد سوف يحتوى على كل المعلومات الموجودة في ملف البيانات الأصل ، علاوة على درجة الاختبار الإضافية ومتوسط الدرجات لكل طالب .

طريقة إجراء الحسابات Computational Procedure

سوف نكتب البرنامج بطريقة تسمح لنا « إذا رغبتنا في ذلك » . بإضافة درجة اختبار جديدة لكل طالب بدون حساب المتوسط وذلك يمكننا من استخدام البرنامج أثناء الفصل الدراسى لتسجيل بيانات إضافية (درجات اختبار) وعند نهاية الفصل الدراسى يسمح بتشغيل هذه البيانات (أى حساب متوسط الدرجات التى تستخدم لتقرر درجة الطالب النهائية) .

وحتى يكون البرنامج برنامجاً عاماً بقدر المستطاع ، دعنا ندخل كل درجات الاختبار للطالب في المجموعة المتراسة C . وبذلك فإن C(1) سوف تشير إلى درجة الاختبار الأولى و C(2) لدرجة الاختبار الثانية و... إلخ . وسوف نسمح بعدد من الدرجات لكل طالب تصل إلى 15 درجة اختبار .

وسوف يشير المتغير K لاختبار معين يتم إدخاله من الكونسول على سبيل المثال . إذا كانت الدرجات التى ندخلها هي للاختبار السادس ، فإننا نحدد قيمة K بعد ذلك بالرقم 6 وكذلك فإن J\$ سوف تمثل إما "YES" أو "NO" ويتوقف ذلك على كون حساب متوسط درجات الطالب قد تم أم لا ، وباقي المتغيرات سوف يكون لها نفس التعاريف كما في المثال ٨ - ٣ .

وسوف تواصل الحسابات كما يلي :

١ - حدد ملف البيانات SCORES قناة رقم 1 ، والملف UPDATE قناة رقم 2 (وسوف يكون SCORES ملف إدخال UPDATE ملف إخراج) .

٢ - ألغ أى بيانات قديمة يمكن أن تظهر على الملف UPDATE وأعد وضع الملف إلى نقطة البداية وذلك إستعداداً للكتابة عليه .

٣ - اقرأ رقم السطر (N) وعنوان المادة الدراسية (T\$) والفصل الدراسى (Y\$) من الملف SCORES .

٤ - اطبع عنوان المادة والفصل الدراسى على الكونسول .

٥ - اطبع رقم السطر وعنوان المادة والفصل الدراسى على UPDATE .

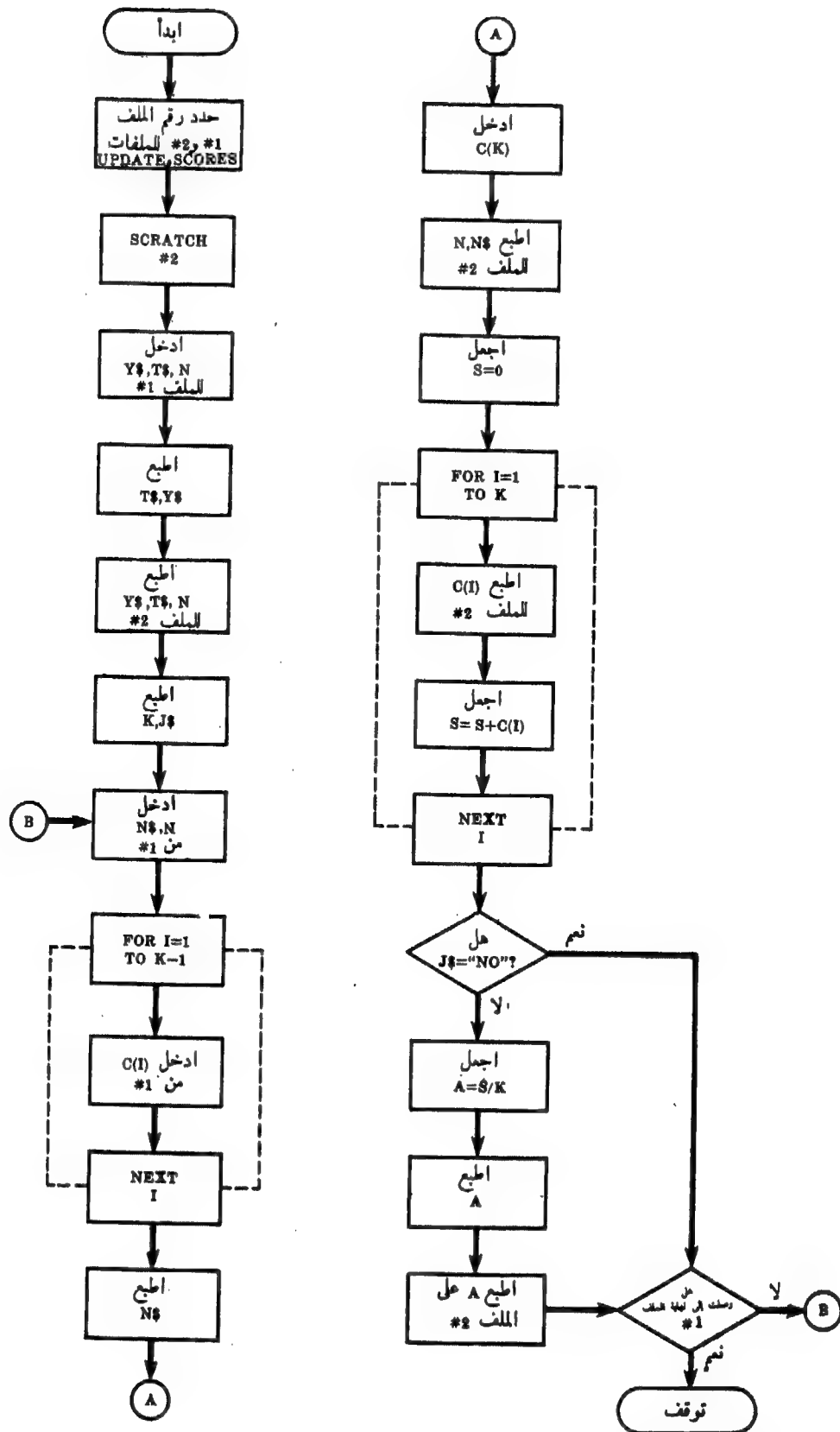
٦ - ادخل قيمة لـ K (رقم الاختبار) من الكونسول .

٧ - أدخل قيمة للمتغير J\$ (أما « نعم » "YES" أو « لا » "NO") من الكونسول ، مبيناً إذا كان المطلوب حساب المتوسط لكل طالب أم لا .

٨ - اقرأ رقم سطر (N) واسم طالب (N\$) وعدد درجات 1 — K من الملف .

٩ - أدخل درجة الاختبار التى رقبها K من الكونسول .

١٠ - اكتب رقم السطر ، واسم الطالب ، ودرجات الطالب كلها وعددها K الملف UPDATE .



١١- إذا كانت $J \neq \text{"NO"}$ تقدم للخطوة ١٣ التالية. وإلا ، احسب متوسط درجات الاختبار (A) باستخدام الصيغة :

$$A = \frac{C(1)+C(2)+\dots+C(K)}{K}$$

١٢- اطبع متوسط درجات الاختبار على النهاية الطرفية المركزية واكتب هذه القيمة أيضاً على الملف UPDATE

١٣- اختبر لترى ما إذا كان هناك بيانات إضافية في الملف SCORES . إذا كان ذلك صحيحاً ، فارجع مرة أخرى للخطوة ٨ ، وإلا أنه الحسابات .

خريطة سير العمليات المناظرة لذلك مبينة في شكل ٨ - ٤ .

برنامج بيسك The BASIC Program

يحتوى شكل ٨ - ٥ على برنامج البيسك الحقيقي . لاحظ تضيئين عدة جمل للتعامل مع الملفات في البرنامج .

تحدد جملة FILES الموجودة في السطر 50 الملفات البيانات SCORES و UPDATE قناتي البيانات 1 و 2 على الترتيب . يجب أن تسبق هذه الجملة أى جملة من جمل الملفات . وبمجرد إعطاء الملفات أرقاماً كقنوات للبيانات المناظرة ، فإن الرجوع لهذه الملفات سوف يكون بواسطة أرقام القنوات وليس بأسماء هذه الملفات .

تحدد الجملة رقم 60 (QUOTE) إن كل السلاسل الحرفية التي تكتب على UPDATE سوف تكون محصورة بين علامتي اقتباس . ويكون ذلك مهماً إذا أردنا قراءة هذه السلاسل الحرفية برنامج بيسك في أى وقت لاحق . يترتب على جملة SCRATCH (سطر رقم 70) مسح أى بيانات سبق تخزينها على الملف UPDATE وإعادته إلى نقطة البداية . ويكون جاهزاً لكتابة أى بيانات .

تقرأ البيانات من الملف SCORES في أماكن متعددة من البرنامج (السطور 80 و 180 و 200) . وبالمثل فإن البيانات تكتب على الملف UPDATE في أماكن متعددة (السطور 100 و 240 و 270 و 330 و 340) . لاحظ أن أرقام السطور تقرأ وتكتب ، بالرغم من عدم استخدامها في البرنامج . ونرى أيضاً أن المتغيرات التي تظهر في جمل PRINT تتبناها الفاصلة المنقوطة (;) . ويترتب على ذلك أن تكون بيانات كل سطر من الملف UPDATE متقاربة بقدر الإمكان .

```

10 REM PROGRAM TO PROCESS STUDENT EXAMINATION SCORES
20 REM USING SEQUENTIAL DATA FILES
30
40 DIM C(15)
50 FILES SCORES, UPDATE
60 QUOTE #2
70 SCRATCH #2
80 INPUT #1,N,T$,Y$
90 PRINT "COURSE:";T$,"TERM:";Y$
100 PRINT #2,N;T$;Y$
110 PRINT
120 PRINT "EXAM NUMBER:";
130 INPUT K
140 PRINT
150 PRINT "CALCULATE AVERAGES (YES OR NO)";
160 INPUT J$
170 PRINT
180 INPUT #1,N,N$
190 FOR I=1 TO K-1
200 INPUT #1,C(I)
210 NEXT I
220 PRINT N$;"SCORE=";
230 INPUT C(K)
240 PRINT #2,N;N$;
250 LET S=0
260 FOR I=1 TO K
270 PRINT #2,C(I);
280 LET S=S+C(I)
290 NEXT I
300 IF J$="NO" THEN 340
310 LET A=S/K
320 PRINT "AVERAGE=";A
330 PRINT #2,A;
340 PRINT #2
350 PRINT
360 IF END #1, THEN 380
370 GOTO 180
380 END

```

وأخيراً نفحص في السطر 360 هل وصلنا لنهاية الملف SCORES ويتحول التحكم للسطر 380 (END) إذا وجدت نهاية الملف ، وبذلك نهي الحسابات . وإلا فإن التحكم يتحول إلى الجملة التي تليها في التنفيذ السطر 370 ، وذلك لتشغيل درجات الطالب التالي .

تطبيق البرنامج Application of the Program

نرى في شكل ٨ - ٦ كيف يمكن استخدام البرنامج لإضافة درجة اختبار إضافية لكل طالب ، وبدون حساب المتوسط . كل درجة اختبار جديدة تدخل من خلال النهاية الطرفية المركزية (استجابات المستخدم موضوع تحتها خط) .

يحتوي ملف البيانات الجديد (UPDATE) درجات اختبارات . ذلك مبين في نهاية شكل ٨ - ٦ . وملف البيانات الجديد يولد من ملف البيانات الأصلي SCORES ، وهو مبين في شكل ٨ - ١ . ويستخدم أمر النظام LIST لإصدار قائمة بملف البيانات UPDATE على الكونسول ، بعد الانتهاء من تنفيذ البرنامج .

COURSE:COMP SCI 100 TERM:FALL, 1982

EXAM NUMBER ?٤

CALCULATE AVERAGES (YES OR NO) ?N

ADAMS B F SCORE= 775
BROWN P SCORE= 780
DAVIS R A SCORE= 735
FISHER E K SCORE= 75
HAMILTON S P SCORE= 790
JONES J J SCORE= 780
LUDWIG C W SCORE= 770
OSBORNE T SCORE= 780
PRINCE W F SCORE= 7100
RICHARDS E N SCORE= 770
SMITH M C SCORE= 775
THOMAS B A SCORE= 710
WOLFE H SCORE= 795
ZORBA D R SCORE= 795

TIME: 0.80 SECS.

>OLD

OLD FILE NAME-->UPDATE

>LIST

UPDATE

10 "COMP SCI 100" "FALL, 1982"
20 "ADAMS B F" 45 80 80 95 55 75
30 "BROWN P" 60 50 70 75 55 80
40 "DAVIS R A" 40 30 10 45 60 55
50 "FISHER E K" 0 5 5 0 10 5
60 "HAMILTON S P" 90 85 100 95 90 90
70 "JONES J J" 95 90 80 95 85 80
80 "LUDWIG C W" 35 50 55 65 45 70
90 "OSBORNE T" 75 60 75 60 70 80
100 "PRINCE W F" 85 75 60 85 90 100
110 "RICHARDS E N" 50 60 50 35 65 70
120 "SMITH M C" 70 60 75 70 55 75
130 "THOMAS B A" 10 25 35 20 30 10
140 "WOLFE H" 25 40 65 75 85 95
150 "ZORBA D R" 65 80 70 100 60 95

>OLD
OLD FILE NAME-->SCORES
>SCRATCH
>OLD
OLD FILE NAME-->UPDATE
>RENAME SCORES

نفرض أننا أردنا أن نستخدم ملف البيانات الجديد في وقت لاحق كلف إدخال لنفس برامج البيسك (BASIC) ولذلك فن المهم أن نغير اسم ملف البيانات من UPDATE إلى SCORES ويمكن إنجاز ذلك باستخدام أوامر النظام SCRATCH (وذلك لإزالة ملف البيانات الأصلي SCORES) ثم الأمر RENAME (وذلك لتغير اسم UPDATE إلى SCORES). والإجراء مبين

شكل ٨ - ٧

في شكل ٨ - ٧ . (لا تخلط أمر نظام البيسك SCRATCH مع جملة التعامل مع الملف SCRATCH ، على سبيل المثال (SCRATCH##2) .

والآن دعنا نرى كيف نستخدم هذا البرنامج لإدخال درجة اختبار جديدة ثم بعد ذلك حساب متوسط الدرجات لكل طالب . درجات الاختبار الجديدة والمتوسطات المحسوبة مبينة في شكل ٨ - ٨ . (هذا المخرج مولد باستخدام ملف البيانات الأصلي SCORES المبين في شكل ٨ - ١) .

لاحظ التماثل بين شكل ٨ - ٨ وشكل ٨ - ٣ (والأخير مولد باستخدام البرنامج في المثال ٨ - ٣) . ولكننا الآن ، ليس عندنا فقط سجل لمتوسط درجات اختبار لكل طالب مبين على النهاية الطرفية المركزية ، ولكن سجل كامل لدرجات الاختبار والمتوسطات في ملف البيانات UPDATE . قائمة بالملف UPDATE مبينة في نهاية الشكل ٨ - ٨ .

ملفات البيانات التسلسلية مفيدة عامة لتخزين مجموعات من البيانات التي سوف تشغل بواسطة برنامج بيسك في أي وقت لاحق . وهناك سببان لذلك . أولاً ، أنه يمكن تنقيح ملف بيانات تسلسل باستخدام أوامر نظام بيسك بدون تغيير البرنامج . وعلاوة على ذلك ، فن الممكن تفمين كل من البيانات الحرفية والرقية في ملف بيانات تسلسل .

ومن الناحية الأخرى ، فبعض التطبيقات تتطلب تحويل المعلومات من وإلى ملف البيانات بدون اعتبار للترتيب الذي تخزن به البيانات . واستخدام ملف بيانات تسلسل غير فعال نسبياً في مثل هذه المواقف ، حيث يمكن تصحيح وقت الحساب الجديد بالاعتبار بالبحث تكرر أراً في ملف البيانات على المعلومات المطلوبة . وفي الجزء التالي سوف ندرس نوعاً مختلفاً من ملفات البيانات أكثر ملاءمة للتطبيقات من النوع التسلسل .

٨ - ٢ ملفات البيانات العشوائية RANDOM DATA FILES

بينما يحتوي ملف البيانات التسلسلية على مجموعات من البيانات مرتبة على أساس تزايد أرقام السطور ، فإن ملف البيانات العشوائية يحتوي على بنود بيانات فردية وغير مرتبة بأي نظام معين . كل بند من بنود البيانات يمكن قراءته أو كتابته مباشرة من أو إلى ملف بيانات عشوائية بدون مواصلة متتالية في ملف البيانات من البداية . ولذلك فإن تحويل المعلومات ومن وإلى ملف البيانات العشوائية أسرع من ملف البيانات التسلسلية .

يمكن أن تتكون ملفات البيانات العشوائية من بيانات رقمية أو حرفية ولكن ليس الاثنان معاً . يوصف نوع الملف (إما رقمي أو حرفي) بواسطة وضع علامة النسبة المئوية (%) أو علامة الدولار (\$) بعد اسم الملف . تسمى علامة % أن الملف رقمي ، وعلامة (\$) على أن الملف حرفي . ويجب أن تتبع علامة (\$) كمية صحيحة موجبة (تتراوح من 1 إلى 132) هذه الكمية تحدد العدد الأقصى من الحروف التي يمكن أن تظهر في كل سلسلة حرفية .

مثال ٨ - ٥

برنامج بيسك يحتوي على الجملة :

10 FILES SALES%,MASTER%,ITEMS\$20,CUSTMR\$72

COURSE: COMP SCI 100 TERM: FALL, 1982

EXAM NUMBER 76

CALCULATE AVERAGES (YES OR NO) ?YES

ADAMS B F SCORE= 775
AVERAGE= 71.6667BROWN P SCORE= 780
AVERAGE= 65DAVIS R A SCORE= 755
AVERAGE= 40FISHER E K SCORE= 75
AVERAGE= 4.16667HAMILTON S P SCORE= 790
AVERAGE= 91.6667JONES J J SCORE= 780
AVERAGE= 87.5LUDWIG C W SCORE= 770
AVERAGE= 53.3333OSBORNE T SCORE= 780
AVERAGE= 70PRINCE W F SCORE= 7100
AVERAGE= 82.5RICHARDS E N SCORE= 770
AVERAGE= 55SMITH M C SCORE= 775
AVERAGE= 67.5THOMAS B A SCORE= 710
AVERAGE= 21.6667WOLFE H SCORE= 795
AVERAGE= 64.1667ZORBA D R SCORE= 795
AVERAGE= 78.3333

TIME: 0.92 SECS.

>OLD

OLD FILE NAME-->UPDATE

>LIST

UPDATE

```

10 "COMP SCI 100" "FALL, 1982"
20 "ADAMS B F" 45 80 80 95 55 75 71.6667
30 "BROWN P" 60 50 70 75 55 80 65
40 "DAVIS R A" 40 30 10 45 60 55 40
50 "FISHER E K" 0 5 5 0 10 5 4.16667
60 "HAMILTON S P" 90 85 100 95 90 90 91.6667
70 "JONES J J" 95 90 80 95 85 80 87.5
80 "LUDWIG C W" 35 50 55 65 45 70 53.3333
90 "OSBORNE T" 75 60 75 60 70 80 70
100 "PRINCE W F" 85 75 60 85 90 100 82.5
110 "RICHARDS E N" 50 60 50 35 65 70 55
120 "SMITH M C" 70 60 75 70 55 75 67.5
130 "THOMAS B A" 10 25 35 20 30 10 21.6667
140 "WOLFE H" 25 40 65 75 85 95 64.1667
150 "ZORBA D R" 65 80 70 100 60 95 78.3333

```

شكل ٨ - أ

تم تحديد قنوات البيانات 1 و 2 للملفات SALES و MASTER وهي ملفات بيانات عشوائية رقمية ، كما تشير إلى ذلك (%) علامة النسبة المئوية التي تتبع اسم كل ملف . تم تحديد قنوات البيانات 3 و 4 للملفات ITEM و CUSTMR وهي ملفات بيانات عشوائية حرفية حيث أقصى طول للسلسلة الحرفية هو 20 و 72 حرفاً على الترتيب .

لاحظ أن علامة (%) النسبة المئوية أو علامة (\$) الدولار يتبعها رقم صحيح موجب ليست جزءاً من اسم الملف الحقيقي . ولكنها ملحق بالاسم (تذكر أن كل اسم من أسماء الملفات بدون الملاحق لا يمكن أن يتعدى 6 حروف) .

وملفات البيانات العشوائية على عكس ملفات البيانات المتتالية ، لا يمكن أن نعمل لها قائمة مباشرة على الكونسول . ولا يمكن أن نتفح باستخدام أوامر نظام البيسك . بينما يمكن بسهولة كتابة برنامج بيسك يقوم بأحد هذه المهام أو كليهما . وسوف ترى كيف يمكن إنجاز ذلك في فصل لاحق . أولاً ، يجب أن ندرس كيف يمكن التوصل لكل بند منفرد من البنود الموجودة في ملف البيانات العشوائي .

تحكم المؤشر Pointer Control

بالرغم من أن بنود البيانات في ملف البيانات العشوائي ليست منظمة بأي ترتيب . إلا أن أماكن بنود البيانات مرقمة بترقيم متتالي من بداية الملف (مبتدئاً بالمكان رقم 1 ويزاد بمقدار الوحدة لكل بند بيانات متعاقب) . ويستخدم مؤشر للإشارة إلى مكان كل بند منفرد من البيانات . ويجب أن يكون المؤشر دائماً موضوعاً في الوضع الصحيح قبل نقل أي بند من البيانات من أو إلى ملف البيانات .

مثال ٨ - ٦

يحتوي ملف بيانات عشوائي على ثمان قيم رقمية منظمة بالترتيب المبين فيما يلي :

الموضع	القيمة
1	433
2	256
3	307
4	180
5	75
6	224
7	609
8	52

نفرض أننا نريد أن نقرأ بنود البيانات الخامس والثاني والسابع بهذا الترتيب . أولاً نضع المؤشر في الموضع رقم 5 ونقرأ القيمة 75 . ثم نعيد وضع المؤشر في الموضع رقم 2 ثم نقرأ القيمة 256 . وأخيراً ، نحرك المؤشر إلى الموضع رقم 7 ونقرأ القيمة 609

وسوف يتقدم المؤشر أوتوماتيكياً خطوة واحدة في كل مرة يقرأ فيها بنوداً أو نكتب بنوداً من الملف . وبذلك فن الممكن أن نقرأ أو نكتب بنوداً متتالية من ملف بيانات عشوائي . ويمكننا إعادة موضع المؤشر وقتها نريد ، وذلك بواسطة جملة SET (وأي بعض نسخ بييسك ، جملة RESET) ، وتسمح لنا هذه الجملة بالتوصل لبنود البيانات بأي ترتيب نريده .

ومقترناً بجملة SET الدالتين المكتبتين LOC و LOF . تسمح لنا دالة LOC بتحديد موضع المؤشر ، وتشير دالة LOF لآخر مكان تخزين في الملف . والمثال التالي يوضح استخدام جملة SET ودالتى LOC و LOF .

مثال ٨ - ٧

فيما يلي بناء هيكل لبرنامج بييسك يتعامل مع بيانات من ملف بيانات عشوائي رقمي .


```

10 FILES VALUES%
...
40 SET :1,K
...
80 IF LOC(1)>LOF(1) THEN 200
...
110 SET :1,LOC(1)+3
...
150 SET :1,(1+LOF(1))/2

```

تحدد جملة 15 قناة البيانات 1 لملف البيانات العشوائى الرقى VALUES . وفى الجملة رقم 40 يحدد موضع المؤشر لقناة البيانات 1 بالمكان المشار إليه بواسطة المتغير K (نفرض أن K لها قيمة صحيحة موجبة) . وتسبب الجملة رقم 80 تحويل التحكم للجملة رقم 200 إذا كان المؤشر الخاص بقناة البيانات 1 موضوعاً فى أى مكان وراء نطاق نهاية الملف .

وفى الجملة رقم 110 نعيد وضع المؤشر لقناة البيانات 1 ثلاثة أماكن وراء نطاق الموضع الحال له . وأخيراً ، فى الجملة 150 نعيد وضع المؤشر إلى نقطة المنتصف فى الملف ، محدداً كتوسط لأول وآخر موقع فى الملف .

لاحظ أن مكان المؤشر فى الجمل 110 و 150 تم تحديدها بواسطة الصيغة $LOC(1) + 3$ و $(1 + LOF(1))/2$ بالترتيب . مسجوح بالصيغة الرياضية فى جملة SET على أن تكون قيمتها موجبة ولكن لا تتعدى آخر موضع فى الملف . وسوف تبتز القيم غير الصحيحة أوتوماتيكياً .

ولاحظ أيضاً ، أن رقم القناة فى جملة SET تسبقها علامة الوقف الاستدراكى (:) بدلا من علامة الجنيه \pounds . وبهذه الطريقة نميز ملف البيانات العشوائى عن ملف البيانات التسلسلى . كل جمل التعامل مع الملفات والتى تشير إلى رقم القناة تستعمل هذه العلامة .

وسوف نرى برنامج ببسك كاملا يستخدم جملة SET والدالتين LOC و LOF فى المثال ٨ - ١٣ .

قراءة ملف بيانات عشوائى Reading a Random Data File

لقد رأينا أن ملف البيانات العشوائى يمكن قراءته على التوالى أو عشوائياً . وإذا أردنا أن نقرأ البيانات على التوالى فيمكن إغفال موضع المؤشر ، حيث أن جملة FILES تضع المؤشر عند أول موضع فى الملف وفى كل مرة يقرأ بند من البيانات يتقدم المؤشر أوتوماتيكياً إلى الموضع التالى مباشرة .

مثال ٨ - ٨

يحتوى ملف البيانات العشوائى STATES على أسماء 50 ولاية فى الولايات المتحدة الأمريكية (U.S.A.) ومرتبطة هجائياً . يمثل شكل ٨ - ٩ برنامج ببسك يقرأ اسم كل ولاية من ملف البيانات ثم يطبع الاسم على الكونسول . وسوف نقرأ البيانات على التوالى وبنفس الترتيب التى تم تخزينها به فى الملف .

تنجز جملة FILES عدة أشياء فى هذا البرنامج . أولاً ، تحدد للملف البيانات العشوائى للسلاسل الحرفية STATES قناة البيانات رقم 1 ، وتصف كل سلسلة حرفية بعدد من الحروف لا يتجاوز 15 حرفاً .

لاحظ أننا نستعمل جملة التعامل مع الملف READ وليس INPUT عند القراءة من ملف البيانات العشوائى . (فى عديد من جمل البيسك تستخدم جمل READ و WRITE لملفات البيانات العشوائية وجمل INPUT و PRINT لملفات البيانات التسلسلية) .

```

10 FILES STATES$15
20 FOR I=1 TO 50
30   READ 1, N$
40   PRINT N$
50 NEXT I
60 END

```

>RUN

FILGEN 18:52 23-MAR

ALABAMA
ALASKA
ARIZONA
ARKANSAS
CALIFORNIA
COLORADO
CONNECTICUT
DELAWARE
FLORIDA
GEORGIA
HAWAII
IDAHO
ILLINOIS
INDIANA
IOWA
KANSAS
KENTUCKY
LOUISIANA
MAINE
MARYLAND
MASSACHUSETTS
MICHIGAN
MINNESOTA
MISSISSIPPI
MISSOURI
MONTANA
NEBRASKA
NEVADA
NEW HAMPSHIRE
NEW JERSEY
NEW MEXICO
NEW YORK
NORTH CAROLINA
NORTH DAKOTA
OHIO
OKLAHOMA
OREGON
PENNSYLVANIA
RHODE ISLAND
SOUTH CAROLINA
SOUTH DAKOTA
TENNESSEE
TEXAS
UTAH
VERMONT
VIRGINIA
WASHINGTON
WEST VIRGINIA
WISCONSIN
WYOMING

شكل ٨ - ٩

TIME: 0.08 SECS.

الخرج الناتج من هذا البرنامج (أى قائمة بملف البيانات) مبين عند نهاية شكل ٨ - ٩ .
عند قراءة البيانات الموجودة في الملف عشوائياً . يجب أن نحرك المؤشر إلى المكان الصحيح قبل قراءة أى بند . ونستخدم جملة SET لهذا الغرض ، كما هو مبين في المثال التالي — مثال ٨ - ٩ .

مثال ٨ - ٩

دعنا ندرس مرة أخرى ملف البيانات العشوائى STATES ، كما وصفناه في المثال ٨ - ٨ . ونرغب الآن في كتابة برنامج ييسك يؤدي الخطوات التالية :

١ - يدخل من الكونسول . موضع السلسلة الحرفية في الملف STATES (كل موضع يمكن توصيفه بواسطة رقم صحيح ثابت له قيمة تتراوح ما بين 1 إلى 50) .

٢ - تقرأ السلسلة الحرفية (أى اسم الولاية) الموجودة في هذا الموضع .

٣ - تطبع السلسلة الحرفية على الكونسول .

يتكرر هذا الإجراء حتى يتحدد موضع له قيمة أقل من 1 أو أكبر من 50 .

يمثل الشكل ٨ - ١٠ برنامج البيسك المطلوب . وتمثل المتغيرات L و N\$ الموضع المطلوب والسلسلة الحرفية الموجودة في هذا الموضع ، على الترتيب . لاحظ أن جملة SET تسبق جملة READ ، وبذلك يوضع المؤشر عند الموضع الصحيح قبل قراءة كل بند من البيانات .

```
10 FILES STATES#15
20 PRINT "LOCATION";
30 INPUT L
40 IF L<1 THEN 110
50 IF L>50 THEN 110
60 SET :1,L
70 READ :1,N$
80 PRINT N$
90 PRINT
100 GO TO 20
110 END
```

>RUN

EXB.9 19:02 23-MAR

LOCATION 733
PENNSYLVANIA

LOCATION 723
MINNESOTA

LOCATION 750
WYOMING

LOCATION 711
HAWAII

LOCATION 78
DELAWARE

LOCATION 745
VERMONT

LOCATION 70

TIME: 0.31 SECS.

شكل ٨ - ١٠

يبين الجزء الأسفل من شكل ٨ - ١٠ المخرج الذي يولد كإجابة لإدخال عدة كميات . لاحظ أن أسماء الولايات تطبع بالترتيب الموصوف بواسطة جملة INPUT وليس بالترتيب التي تخزن به الأسماء في ملف البيانات .

كتابة ملف بيانات عشوائى Writing a Random Data File

يمكن كتابة بند بيانات على ملف عشوائى بطريقة تشبه طريقة القراءة ، غير أننا نستخدم الآن جملة التعامل مع الملف WRITE بدلاً من READ . وكما ذكرنا سابقاً ، فإن المؤشر يجب وضعه في المكان الصحيح قبل كتابة بند البيانات . وسوف يحمل بند البيانات الجديد محل ما كان مخزن سابقاً في هذا الموضع .

مثال ٨ - ١٠ مراقبة المخزون Inventory Control

يحتفظ مستودع بقائمة جرد لعدد معين من السلع . وسوف تتذبذب كمية كل سلعة من يوم إلى آخر . وذلك تبعاً للأوامر التي تطلب بواسطة العملاء (وبذلك تقل كمية المخزون) وتبعاً لوصول الشحنات من الموردين (وبذلك تزيد كمية المخزون) . ونريد الاحتفاظ بسجل بالمخزون المضبوط لكل بند . ولعمل ذلك يجب أن نعدل مستوى المخزون بعد كل معاملة على حدة .

طريقة إجراء الحسابات Computational Procedure

يمكن تسجيل مستوى المخزون لكل سلعة بطريقة ملائمة في ملف بيانات عشوائى رقمى . وسوف نستعمل ملف البيانات INVTRY لهذا الغرض .

سوف نحدد كل سلعة برقم مخزون فريد (رقم صحيح موجب) يعرف بالسلعة . وسوف يستخدم رقم المخزون أيضاً ليشير إلى موضع سجل المخزون في ملف البيانات . وبذلك إذا كان المطلوب معرفة كم عدد الوحدات للسلعة رقم 86 الموجودة حالياً في المستودع فيجب أن نفحص محتويات مكان التخزين رقم 86 في ملف البيانات .

- ١ - حدد ملف البيانات INVTRY بقناة البيانات 1 .
- ٢ - اطبع حجم ملف البيانات (أى قيمة آخر موضع تخزين) على الكونسول .
- ٣ - ادخل قيمة لرقم المخزون (P) من على الكونسول .
- ٤ - إذا كانت قيمة P أقل من 1 أو أكبر من آخر موضع في ملف البيانات ، أنه البرنامج والحسابات ، وألا تقدم للخطوة ٥ التالية .
- ٥ - ضع مؤشر ملف البيانات في الموضع P .
- ٦ - اقرأ مستوى المخزون (N) من ملف البيانات .
- ٧ - اطبع القيمة الحالية N على الكونسول .
- ٨ - أدخل التغير في مستوى المخزون (N1) ، بدلالة عدد الوحدات من الكونسول . (لاحظ نقص مستوى المخزون يجب أن يشار إليه بإدخال قيمة سالبة للمتغير N1)
- ٩ - احسب مستوى مخزون جديد ، أى ، اجمل :

$$N = N + N1$$

١٠ - إذا كانت قيمة N الجديدة سالبة (وهذا لا يتحقق في الواقع) فاجمل قيمة N مساوية للصفر .

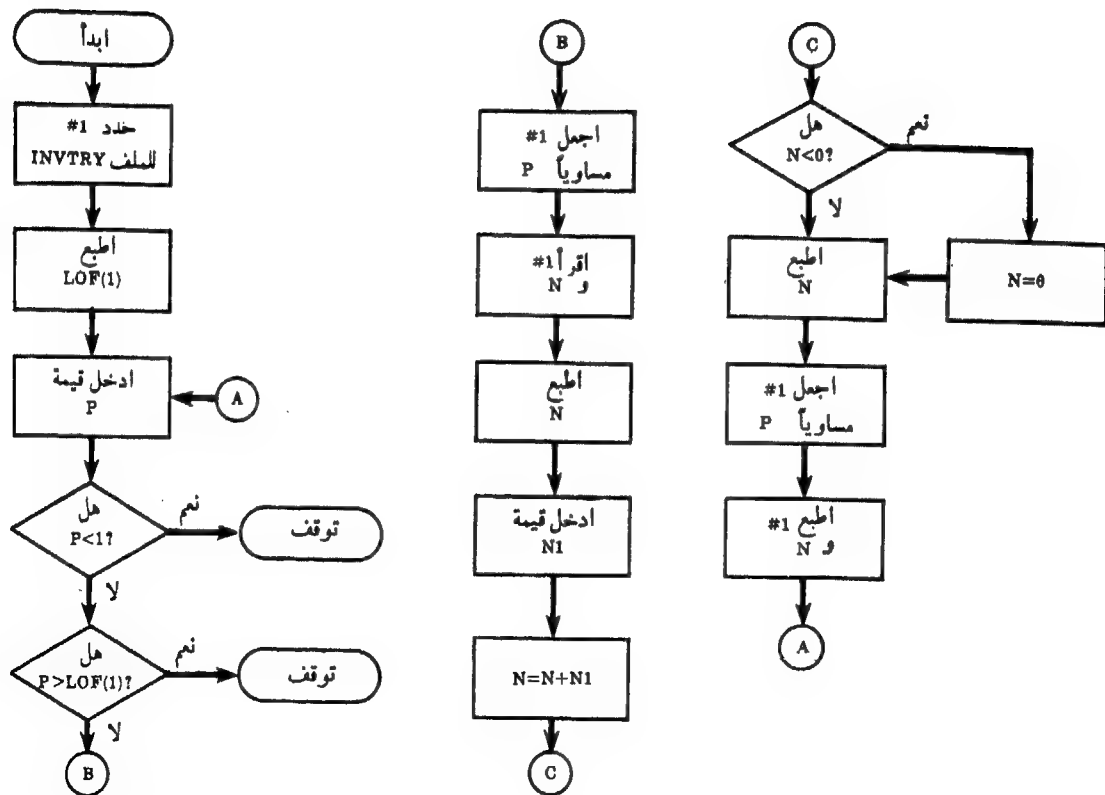
١١ - اطبع القيمة الجديدة N على الكونسول .

١٢ - أعد المؤشر للموقع P (لاحظ أن المؤشر سوف يتقدم للموقع P + 1 بعد الخطوة رقم ٦) .

١٣ - اكتب قيمة N الجديدة على ملف البيانات .

١٤ - ارجع مرة ثانية للخطوة ٣

يبين شكل ٨ - ١١ خريطة سير العمليات المناظرة لذلك .



شكل ٨ - ١١

```

10 REM INVENTORY CONTROL PROGRAM
20 FILES INVTYX
30 PRINT "STOCK NUMBERS RUN FROM 1 TO ",LOF(1)
40 PRINT
50 PRINT "STOCK NUMBER";
60 INPUT P
70 IF P<1 THEN 220
80 PRINT
90 SET :1,P
100 READ :1,N
110 PRINT "ORIGINAL INVENTORY=";N;" ITEMS"
120 PRINT "CHANGE IN INVENTORY LEVEL";
130 INPUT N1
140 LET N=N+N1
150 IF N>=0 THEN 170
160 LET N=0
170 PRINT "NEW INVENTORY=";N;" ITEMS"
180 PRINT
190 SET :1,P
200 WRITE :1,N
210 GOTO 50
220 END

```

>RUN

EXB.10 19:04 23-MAR

STOCK NUMBERS RUN FROM 1 TO 2000

STOCK NUMBER ?1186

ORIGINAL INVENTORY= 346 ITEMS
 CHANGE IN INVENTORY LEVEL ?-43
 NEW INVENTORY= 301 ITEMS

STOCK NUMBER ?708

ORIGINAL INVENTORY= 346 ITEMS
 CHANGE IN INVENTORY LEVEL ?200
 NEW INVENTORY= 546 ITEMS

STOCK NUMBER ?84

ORIGINAL INVENTORY= 147 ITEMS
 CHANGE IN INVENTORY LEVEL ?16
 NEW INVENTORY= 163 ITEMS

STOCK NUMBER ?1400

ORIGINAL INVENTORY= 78 ITEMS
 CHANGE IN INVENTORY LEVEL ?-30
 NEW INVENTORY= 28 ITEMS

STOCK NUMBER ?0

TIME: 0.29 SECS.

شكل ٨ - ١٢

برنامج البيسك The BASIC Program

يحتوى شكل ٨ - ١٢ على برنامج بيسك حقيقى . نرى أن البرنامج يحتوى على جمل التعامل مع الملفات READ و SET و FILES و WRITE كذلك الدالة المكتوبة LOF . لاحظ أن أى جملة من جمل READ و WRITE دائماً مسبقة بجملة SET ، وبذلك يتم وضع المؤشر فى المكان الصحيح قبل نقل أى معلومات من أو إلى ملف البيانات .

الجزء الأسفل من شكل ٨ - ١٢ يبين مجموعة بيانات نموذجية ناتجة من تنفيذ البرنامج كذلك يبين كلا من بيانات الإدخال والمخرجات . لاحظ أن تنفيذ البرنامج يتوقف بإدخال قيمة صفر لأحد أرقام المخزون .

يمكننا تصميم عدة بنود للإخراج فى جملة WRITE واحدة إذا أردنا ذلك . فى مثل هذه الحالات فإن أول بند سوف يوضع فى المكان الذى يحدده المؤشر ، وبنود البيانات التالية سوف تخزن فى المواضع المتتالية . كما فى جملة PRINT ، يمكن أن يتمثل كل بند من البيانات بواسطة رقم ثابت أو متغير أو صيغة رياضية أو مرجع دالة .

مثال ٨ - ١١

برنامج بيسك يحتوى على الجمل :

```
10 FILES DATA%
...
100 SET :1,P
110 WRITE :1,C1,(A+B)/2,SQR(X)
```

نفرض أن P قيمتها 39 فإن القيمة التى تعطى للمتغير C1 سوف توضع فى المكان 39 فى ملف البيانات الرقمية DATA وسوف يحتوى الموضع رقم 40 على القيمة الممثلة بالصيغة الرياضية $(A+B)/2$ والموضع رقم 41 على القيمة SQR (X) .

إنشاء ملف بيانات عشوائى Creating a Random Data File

لا يمكن أن ينشأ ملف بيانات عشوائى باستخدام أوامر نظام بيسك ويجب علينا إذن أن نكتب برنامج بيسك لإنشاء ملف بيانات عشوائى . والإجراء لتنفيذ ذلك ممثل فى البرنامج التالى .

مثال ٨ - ١٢

نفرض أننا نرغب فى إنشاء ملف البيانات العشوائى STATES بواسطة برنامج بيسك يسمى FILGEN . برنامج البيسك وأوامر النظام المترجمة معه مبينة فى شكل ٨ - ١٣ . ومرة أخرى فإن أوامر النظام المضافة بواسطة المستخدم كلها موضوع تحتها خط .

لاحظ أن السطور الثلاثة الأولى تعرف الملف الذى تم الاحتفاظ به STATES ، بالرغم من أن البيانات مازالت تدخل إلى STATES وهذا مطلوب حتى يتعرف النظام على جملة FILES التى تعرف اسم الملف (السطر 10)

```
>NEW
NEW FILE NAME-->STATES
>SAVE
>NEW
NEW FILE NAME-->FILGEN
>10 FILES STATES%IS
>20 INPUT N$
>30 IF N$="END" THEN GO
>40 WRITE :1,N$
>50 GO TO 20
>60 END
>SAVE
```

وعند تنفيذ البرنامج FILGEN سوف تدخل مجموعة من السلاسل الحرفية من على الكونسول ونكتبها على STATES ، مبتدئاً بالمكان 1 ويكمل فى مواضع التخزين المتعاقبة وسوف نهى الحسابات عند طباعة كلمة END بعد دخول كل البيانات .

شكل ٨ - ١٣

٨ - ٣ مواصفات ملف اثناء وقت التشغيل RUN TIME FILE SPECIFICATIONS

في عدة تطبيقات نرغب في كتابة برنامج ببسك يستخدم ملفات بيانات ولكنه لا يحدد أى أسماء ملفات معينة . ولكن يمكن أن ندخل أسماء الملفات كبيانات إدخال أثناء تنفيذ البرنامج والبرنامج الذى يكتب بهذه الطريقة سوف يكون برنامجاً عاماً أكثر من برنامج يتطلب ملفات بيانات محددة .

من السهل جداً أن ندخل أسماء الملفات المطلوبة أثناء وقت التشغيل (أى أثناء تنفيذ البرنامج) إذا أردنا ذلك . ويجب في هذه الحالة استخدام جملة FILE وليس جملة FILES (لاحظ أن جملة FILE و FILES مختلفتان تماماً ، سوف ترى ذلك في المثال التالى) .

مثال ٨ - ١٣ البحث في ملف بيانات Searching a Data File

هذا المثال يقدم طريقة فنية فعالة للتوصل لموضع بند بيانات معين في ملف بيانات عشوائى يحتوى على سلاسل الحروف . وسوف نفترض أن سلاسل الحروف مخزنة في الملف بترتيب هجائى . وتعرف الطريقة باسم البحث الثنائى ، وهى تشبه للشكل الممثل في المثال ٦ - ٦ لإيجاد أقصى قيمة للذالة .

طريقة إجراء الحسابات Computational Procedure

دعنا ندرس نطاقاً للبحث يتكون من عدة مواضع تخزين متعاقبة في الملف . مبدئياً سوف يكون نطاق البحث عبارة عن الملف بأكمله . وسوف تكون خطتنا هى مقارنة سلسلة الحروف عند منتصف نطاق البحث مع سلسلة الحروف المطلوبة . سوف نحصل على إحدى ثلاث نتائج .

١ - أن تكون سلسلة الحروف الموجودة عند نقطة المنتصف هى سلسلة الحروف المطلوبة . وفي هذه الحالة سوف يتوقف إجراء الحساب .

٢ - أن تكون سلسلة الحروف المطلوبة في النصف الأول من نطاق البحث ومن ثم يهمل النصف الثانى من نطاق البحث ، وتقارن سلسلة الحروف المطلوبة بسلسلة الحروف عند منتصف النصف الأول من نطاق البحث .

٣ - أن تكون سلسلة الحروف في النصف الثانى من نطاق البحث . وفي هذه الحالة سوف نلغى النصف الأول من نطاق البحث ، ثم تقارن سلسلة الحروف المطلوبة بسلسلة الحروف الموضوعة في منتصف الجزء الفرعى المتبقى من النطاق .

يكرر هذا الإجراء حتى يتم العثور على سلسلة الحروف المطلوبة أو نقرر أن السلسلة غير موجودة في ملف البيانات .

المخطط التمهيدى للبرنامج The Program Outline

من أجل تخطيط الإجراء دعنا نعرف المتغيرات التالية :

F\$ = اسم ملف البيانات العشوائى الحرفى ، متضمناً الملحق .

N\$ = سلسلة الحروف التى يجب وضعها في ملف البيانات .

M\$ = سلسلة الحروف تقرأ من ملف البيانات وتقارن به N\$.

P1 = مؤشر يشير إلى بداية نطاق البحث .

P2 = مؤشر يشير إلى نقطة منتصف نطاق البحث .

P3 = مؤشر يشير إلى نهاية نطاق البحث .

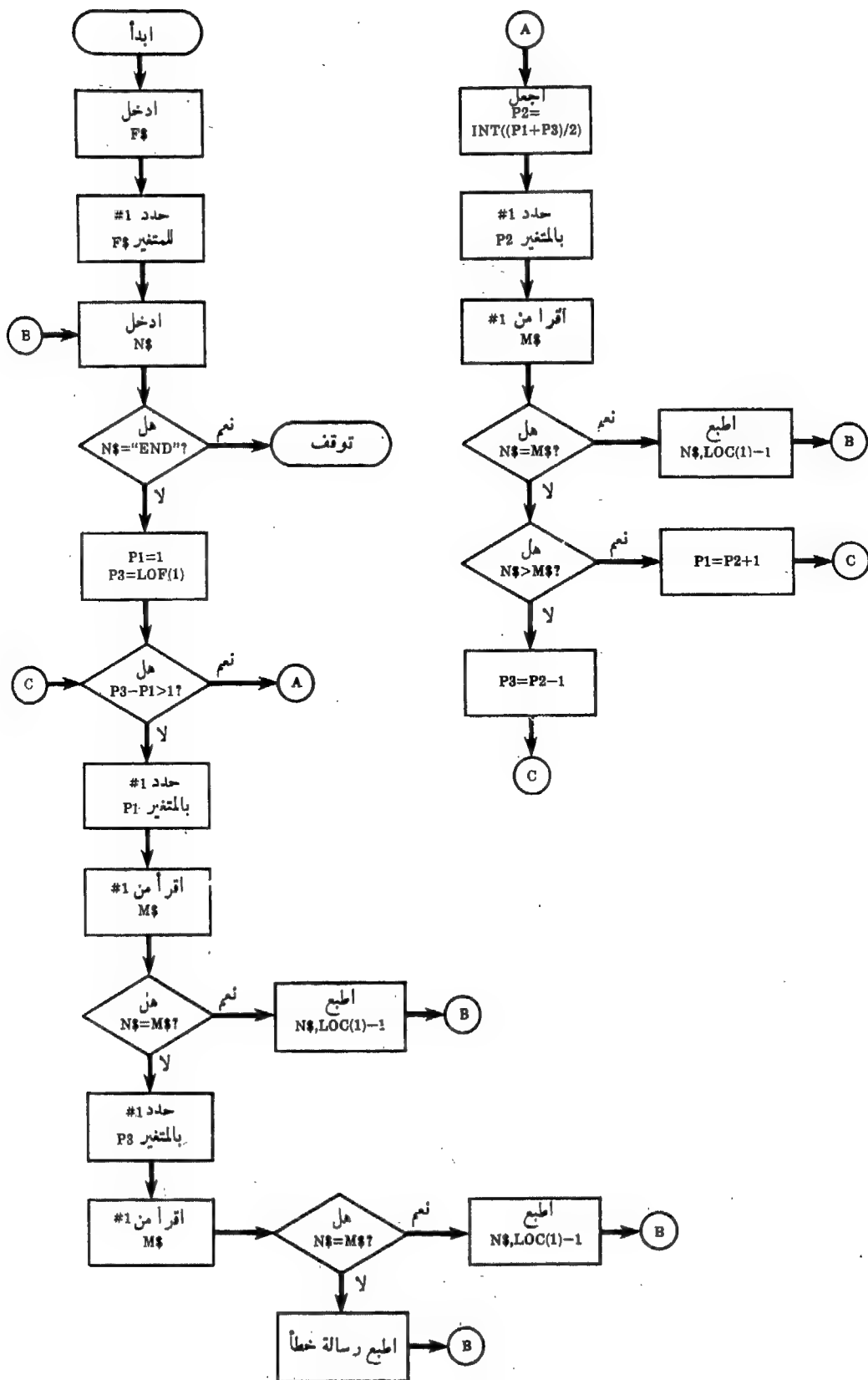
وسوف نواصل الحسابات كما يلي :

- ١ - يدخل اسم الملف من الكونسول وتعطى للمتغير F\$
 - ٢ - يلازم الملف الممثل بواسطة F\$ قناة البيانات رقم 1 .
 - ٣ - تدخل سلسلة الحروف من الكونسول وتعطى للمتغير N\$. إذا كانت N\$=END ، فإن تنفيذ البرنامج سوف يتوقف ، وإلا فسوف نستمر في الحسابات من الخطوة ٤ التالية .
 - ٤ - نحدد قيم مبدئية مقدارها 1 للمؤشرات P1 و P2 و LOF(1) على الترتيب وذلك يعرف نطاق البحث المبدئي .
 - ٥ - إذا ضيقنا نطاق البحث إلى حد أن P1 و P3 تشير إلى مواضع متجاورة ، فإن قيمة M\$ تقرأ من الموضع P1 ثم تقارن بـ N\$
 - (أ) إذا كانت N\$=M\$ فإن التحكم سوف ينتقل إلى الخطوة ٨ أدناه .
 - (ب) إذا كانت M\$ تختلف عن N\$ فتقرأ قيمة أخرى M\$ من الموضع P3 ثم تقارن بـ N\$.
 - (ج) إذا كانت N\$=M\$ فإن التحكم سوف ينتقل إلى الخطوة ٨ أدناه .
 - (د) إذا كانت قيمة M\$ لا تساوى N\$ ، فتقطع رسالة على الكونسول تشير إلى أن السلسلة الحرفية المطلوبة غير موجودة . ثم نرجع مرة ثانية للخطوة ٣ عليه .
 - ٦ - إذا لم تشر P1 و P3 إلى مواضع متجاورة ، بذلك يمكن تحديد قيمة المتغير P2 باستخدام الصيغة الرياضية .

$$P2=INT((P1+P3)/2)$$
 - ٧ - تقرأ قيمة للمتغير M\$ من الموضع P2 ثم يقارن بـ N\$
 - (أ) إذا كانت N\$=M\$ ، فالتحكم سوف ينتقل للخطوة ٨ أدناه .
 - (ب) إذا كانت N\$<M\$ ، نحتفظ بنصف نطاق البحث الأول . من ثم تعطى P3 قيمة جديدة تحسب كالتالى

$$P3=P2-1$$
 ويتحول التحكم مرة ثانية إلى الخطوة رقم ٥ عليه .
 - (ج) إذا كانت N\$>M\$ ، نحتفظ بنصف نطاق البحث الثانى . من ثم نحسب قيمة جديدة للمتغير P1 كالتالى :

$$P1=P2+1$$
 ويتحول التحكم مرة ثانية إلى الخطوة رقم ٥ عليه .
 - ٨ - تطبع رسالة على النهاية الطرفية المركزية تشير إلى أن سلسلة الحروف المطلوبة قد تم تخزينها في الموضع رقم 1 — LOC(1) (لاحظ أننا استخدمنا الموضع رقم 1 — LOC(1) وليس LOC(1) لأن المؤشر سوف يكون قد تقدم خطوة عند قراءة أحدث قيمة للمتغير M\$) .
 - ٩ - بعد ذلك نرجع للخطوة رقم (٣) . وبذلك نعيد البحث عن سلسلة الحروف الجديدة .
- خريطة سير العمليات المناظرة لذلك مبينة في شكل ٨ - ١٤ .



برنامج الفيسك The BASIC Program

يحتوى شكل ٨ - ١٥ على برنامج فيسك كامل لهذه المسألة . ولقد تضمن البرنامج استخدام ثلاث جمل للتعامل مع الملف - وهى FILE (فى السطر 40) و SET (فى السطرين 180 و 290) و READ (السطور 190 و 210 و 300) . وتوجد فى البرنامج أيضاً جملتا LOC و LOF فى السطور 460 و 130 على الترتيب . لاحظ أن جملة FILES المعتادة والتي تحدد قنوات بيانات للملفات البيانات ، غير موجودة . فى الواقع ، فالبرنامج يقرأ اسم ملف فى النهاية الطرفية المركزية (سطر 30) ، ثم بعد ذلك يحدد لاسم الملف هذا قناة البيانات 1 بواسطة جملة FILE فى السطر 40 .

```

10 REM          BINARY SEARCH PROCEDURE
20 PRINT "FILE NAME";
30 INPUT F$
40 FILE :1,F$
50
60 REM          ENTER STRING AND ESTABLISH INITIAL SEARCH INTERVAL
70
80 PRINT
90 PRINT "DESIRED STRING";
100 INPUT N$
110 IF N$="END" THEN 480
120 LET P1=1
130 LET P3=LOF(1)
140
150 REM          TEST FOR SMALL INTERVAL
160
170 IF P3-P1>1 THEN 260
180 SET :1,P1
190 READ :1,M$
200 IF N$=M$ THEN 440
210 READ :1,M$
220 IF N$=M$ THEN 440
230 PRINT N$;" IS NOT IN THE DATA FILE"
240 GOTO 60
250
260 REM          LOCATE MIDPOINT AND COMPARE
270
280 LET P2=INT((P1+P3)/2)
290 SET :1,P2
300 READ :1,M$
310 IF N$=M$ THEN 440
320 IF N$>M$ THEN 390
330
340 REM          RETAIN FIRST HALF OF SEARCH INTERVAL
350
360 LET P3=P2-1
370 GOTO 150
380
390 REM          RETAIN LAST HALF OF SEARCH INTERVAL
400
410 LET P1=P2+1
420 GOTO 150
430
440 REM          DESIRED STRING HAS BEEN LOCATED - PRINT OUTPUT
450
460 PRINT N$;" IS STORED IN LOCATION";LOC(1)-1
470 GOTO 60
480 END

```

شكل ٨ - ١٥

الحوار الناتج من تشغيل هذا البرنامج مبين فى شكل ٨ - ١٦ . ونرى أن ملف البيانات STATES والذي تمت مناقشته فى الأمثلة ٨ - ٨ و ٨ - ٩ و ٨ - ١٢ ، هو الذى نرغب فى فحصه فى هذا المثال . لاحظ أن الملحق (أى \$15) يتم إدخاله مع اسم الملف (البيانات المدخلة موضوع تحتها خط) .

```

FILE NAME ?STATES:13
DESIRED STRING ?PENNSYLVANIA
PENNSYLVANIA IS STORED IN LOCATION 38
DESIRED STRING ?FLORIDA
FLORIDA IS STORED IN LOCATION 9
DESIRED STRING ?OHIO
OHIO IS STORED IN LOCATION 35
DESIRED STRING ?ALASKA
ALASKA IS STORED IN LOCATION 2
DESIRED STRING ?PUERTO RICO
PUERTO RICO IS NOT IN THE DATA FILE
DESIRED STRING ?CALIFORNIA
CALIFORNIA IS STORED IN LOCATION 5
DESIRED STRING ?MASSACHUSETTS
MASSACHUSETTS IS STORED IN LOCATION 21
DESIRED STRING ?END
TIME: 0.45 SECS.

```

شكل ٨ - ١٦

وأخيراً ، يجب أن نؤكد أن هذا البرنامج ، وعلى عكس البرامج التي تمت مناقشتها في الأمثلة السابقة ، يمكن تشغيله مع أي ملف بيانات حرفي - لأن ملف البيانات المقصود فحصه يوصف ككمية مدخلة وليس كجزء من البرنامج . وهذه الطريقة في توصيف الملف تزيد من تعميم البرنامج بدرجة كبيرة .

لم نناقش في هذا الفصل بعض جمل التعامل مع الملف ، مثل RESTORE و MARGIN و PAGE . (بعض نسخ بيسك تتضمن أيضاً مجموعة جمل التعامل مع ملف المصفوفة). والسبب في ذلك ، كما ذكرنا من قبل ، هو التنيرات الموجودة في جمل التعامل مع الملف بين نسخة بيسك وأخرى . والقارئ الذي يرغب في استخدام ملفات بيانات يجب أن يقرر تماماً طبيعة جمل التعامل مع الملفات المتاحة على الجهاز الخاص به .

اسئلة للمراجعة

Review Questions

- ٨ - ١ ما هو الملف ؟ ما هي أنواع المعلومات التي يمكن أن يحتويها ملف ؟
- ٨ - ٢ ما هو الفرق بين ملف بيانات متسلسل وملف بيانات عشوائي ؟
- ٨ - ٣ ما هي مزايا ملف بيانات متسلسل عند مقارنته بملف بيانات عشوائي ؟
- ٨ - ٤ ما هي مزايا ملف بيانات عشوائي عند مقارنته بملف بيانات متسلسل ؟
- ٨ - ٥ كيف ترتب محاميع البيانات في ملف بيانات عشوائي ؟
- ٨ - ٦ هل يمكن دمج الأرقام وسلاسل الحروف في ملف بيانات متسلسل ؟ ملف بيانات عشوائي ؟
- ٨ - ٧ كيف يمكن فصل بنود البيانات الموجودة في ملف بيانات متسلسل عن بعضها ؟ وما هي القواعد الخاصة التي يمكن تطبيقها على سلاسل الحروف فاصلة (,) أو مكان خال ؟
- ٨ - ٨ هل يمكن لأوامر نظام بيسك أن تستخدم في إنشاء وتنقيح ملف بيانات متسلسل ؟ ملف بيانات عشوائي ؟
- ٨ - ٩ ما هي القواعد التي تطبق عند تسمية ملف بيانات ؟
- ٨ - ١٠ لأي نوع من الملفات يجب أن يتبع اسم الملف ملحق ؟ وما هي المعلومات التي يمدنا بها الملحق ؟

- ٨ - ١١ هل يجب أن تقرأ بنود البيانات الموجودة في ملف بيانات متسلسل بترتيب معين ؟
- ٨ - ١٢ كيف يمكن اختيار نهاية ملف بيانات متسلسل ؟
- ٨ - ١٣ ما هي قناة البيانات ؟ وكيف يمكن تحديد قناة البيانات للملف البيانات ؟
- ٨ - ١٤ أين يجب كتابة معلومة جديدة على ملف بيانات عشوائي ؟
- ٨ - ١٥ كيف يمكن مسح البيانات الموجودة على ملف بيانات متسلسل وإعادةه إلى نقطة بدايته ؟
- ٨ - ١٦ كيف يمكن التحكم في التباعد بين بنود البيانات الفردية ، عند كتابة مجموعة من البيانات على ملف بيانات متسلسل ؟
- ٨ - ١٧ كيف يمكن إعادة تسمية ملف بيانات متسلسل ؟
- ٨ - ١٨ كيف يمكن تنظيم بنود البيانات في ملف بيانات عشوائي ؟ وكيف يمكن التوصل إلى بند معين من بنود البيانات ؟
- ٨ - ١٩ ما هو المؤشر ؟ وكيف يمكن إنشاء مكان المؤشر ؟ وكيف يمكن إعادة تحديد وضعه ؟
- ٨ - ٢٠ كيف تقرأ بنود البيانات من ملف بيانات عشوائي ؟ وهل يمكن قراءة ملف بيانات عشوائي بطريقة متتالية ؟ اشرح .
- ٨ - ٢١ كيف تكتب بنود البيانات على ملف بيانات عشوائي ؟ هل يمكن كتابة بيانات على ملف بيانات عشوائي بطريقة متتالية ؟ اشرح .
- ٨ - ٢٢ عند كتابة بند من بنود البيانات على ملف بيانات عشوائي ، ماذا يحدث للمعلومات التي سبق تخزينها في هذا الموضع ؟
- ٨ - ٢٣ كيف يمكن إنشاء ملف بيانات عشوائي ؟
- ٨ - ٢٤ كيف يمكن كتابة برنامج حيث يوصف اسم ملف البيانات أثناء وقت التشغيل ؟ وما هي جمل التعامل مع البيانات التي يجب استخدامها من أجل إنجاز ذلك ؟
- ٨ - ٢٥ عند توصيف ملف البيانات العشوائي أثناء وقت التشغيل ، هل يجب أن نضمن الملحق في اسم الملف ؟
- ٨ - ٢٦ نلخص الغرض من كل من جمل التعامل مع الملفات التالية : FILES و FILE و INPUT و READ و PRINT و WRITE و IF END و QUOTE و SCRATCH و SET
- ٨ - ٢٧ كيف تختلف جمل التعامل مع الملفات INPUT و PRINT عن READ و WRITE ؟
- ٨ - ٢٨ كيف تختلف جمل التعامل مع الملف FILES و FILE عن بعضها ؟
- ٨ - ٢٩ ما هو الغرض من الدوال المكتبية LOC و LOF ؟

مسائل محلولة

Solved Problems

- ٨ - ٣٠ عدة جمل ييسر أو مجموعات من الجمل ، مبينة فيما يلي ، بعض منها مكتوب بصورة غير صحيحة . تعرف على كل الأخطاء .

اسم الملف لا يمكن أن يتعدى الحروف الستة في معظم أنظمة بيسك .

50 PRINT #3,N\$,N\$,P+Q,LOG(T) (ب)

صحيفة ، إذا تم تحديد قناة البيانات 3 لملف البيانات المتسلسل

25 READ :2,N\$,M\$,C1,C2 (ج)

لا يمكن أن يحتوي ملف عشوائي على كل من الثوابت الرقمية والحرفية معا .

150 IF END #1, THEN STOP (د)

يجب أن يستأخر عن كلمة STOP برقم جملة .

10 FILES DATA1%,DATA2% (هـ)

40 READ :1,C1,C2

80 WRITE :2,C1,C2

صحيفة

10 FILES SALES (و)

75 SET #1,P

80 INPUT #1,A,B,T\$,G

تستخدم جملة SET مع ملفات البيانات العشوائية فقط .

60 IF P=LOF(2) THEN 175 (ز)

صحيفة إذا تم تحديد قناة البيانات 2 لملف البيانات العشوائي .

٨-٣١ اكتب جملة أو أكثر من جمل بيسك أو أوامر النظام لكل من المواقف الموصوفة فيما يلي :

(أ) أنشئ ملف بيانات متسلسل SALES احتفظ بالملف ثم أصدر قائمة به بعد أن يتم إدخاله .

```
NEW OR OLD-->NEW
NEW FILE NAME-->SALES
10 ...
20 ...
...
200 ...
SAVE
LIST
```

Data file SALES

(ب) حدد قناة البيانات 1 لملف البيانات المتسلسل SALES

10 FILES SALES

(ج) حدد قناة البيانات 1 لملف البيانات المتسلسل الممثل بالمتغير F\$

10 INPUT F\$

20 FILE #1,F\$

(د) يتكون كل سطر من ملف البيانات المتسلسل FILE 1 من رقم السطر ، تتبعه قيم المتغيرات A و B و P\$ و Q\$ ونرغب في إيجاد قيمة C لكل قيمة من قيم A و B حيث :

$$C = \text{SQR}(A * B)$$

تمت كتابة القيم A و B و C و P\$ و Q\$ على ملف بيانات متسلسل آخر يسمى FILES بفرض أن مجاميع البيانات تقرأ من القناة رقم 2 وتكتب على القناة رقم 4 .

```
10 FILES,FILE1,,FILE2
20 QUOTE #4
30 SCRATCH #4
40 INPUT #2,N,A,B,P$,Q$
50 LET C=SQR(A*B)
60 PRINT #4,N,A,B,C,P$,Q$
70 IF END #2, THEN 90
80 GO TO 40
90 END
```

(هـ) يقرأ برنامج بيانات من ملف البيانات المتسلسل MASTER ثم تكتب البيانات المعدلة على ملف البيانات المتسلسل REVISE . وفي وقت لاحق نرغب في استخدام هذا البرنامج لقراءة البيانات من الملف REVISE وحيث أن البرنامج لن يتغير ، فإننا نرغب في تغيير الاسم REVISE (ملف المخرج الجديد) إلى MASTER (ملف الإدخال القديم) . بين كيف يمكن إنجاز ذلك .

```
OLD
OLD FILE NAME--> MASTER
SCRATCH
OLD
OLD FILE NAME--> REVISE
RENAME MASTER
```

(و) سوف يكتب برنامج بيانات على ملف عشوائى MASTER . بين كيف يتم تعريف ملف خال يسمى MASTER ثم الاحتفاظ به لتشغيل البرنامج .

```
NEW
NEW FILE NAME--> MASTER
SAVE
```

(ز) حدد الموقع الحالى لمؤشر ملف بيانات عشوائى على القناة 3 . حول التحكم إلى نهاية البرنامج إذا وضع المؤشر عند آخر مكان في ملف البيانات .

```
100 LET P=LOC(3)
110 IF P=LOF(3) THEN 250
...
250 END
```

(ح) اقرأ كمية صحيحة موجبة من الكونسول . ثم ضع المؤشر الخاص بقناة البيانات رقم 5 إلى المكان المشار إليه بواسطة الكمية المدخلة .

```
510 INPUT P5
160 SET :5,P5
```

مسائل تكميلية

Supplementary Problems

٨ - ٣٢ قرر أيًا من جمل التعامل مع الملف التالية متاحاً على الجهاز الخاص بك : READ و INPUT و FILE و FILES و PRINT و WRITE و IF END و QUOTE و SCRATCH و SET (أو RESET) و RESTORE و MARGIN و PAGE . هل توجد أي جمل للتعامل مع الملف متاحة علاوة على ذلك ؟

٨ - ٣٣ راجع الفرض من كل من جمل التعامل مع الملف المتاحة على الجهاز الخاص بك . لخص القواعد النحوية لكتابة كل جملة .

٨ - ٣٤ عدة جمل بيسك أو مجموعات من الجمل ، مبينة فيما يلي . بعضها مكتوب بطريقة غير صحيحة . تعرف على كل الأخطاء .

- (a) 35 READ #1,N,A,B,C,P\$,Q\$
- (b) 160 WRITE :2,X,Y,X+Y,X-Y,P\$
- (c) 80 SET :2,LOC(1)+2
90 WRITE :2,X1,X2,X3
- (d) 10 FILES NAMES\$20,ACCTS%
- (e) 10 FILES MASTER
20 SCRATCH MASTER
30 QUOTE MASTER
- (f) 10 INPUT F\$
20 FILES F\$
- (g) 100 SET :1,LOF(1)+3
110 READ :1,L,M,N
- (h) 10 FILES NAMES\$20,ACCTS%
...
75 SET :1,P1
80 READ :1,N\$
85 SET :2,P2
90 WRITE :2,N\$

٨ - ٣٥ اكتب جملة أو أكثر من جمل بيسك أو أوامر النظام لكل من المواقف الموصوفة فيما يلي :

- (أ) حدد قنوات البيانات 1 و 3 وملفات البيانات المتسلسلة LIST 1 و LIST 2 على الترتيب .
- (ب) حدد قناة البيانات 1 وملف البيانات العشوائي الحرفي المسمى NAMES وحدد قناة البيانات 2 وملف البيانات العشوائي الرقمي المسمى ACCTS . افرض أن كل سلسلة الحروف في NAMES سوف تتكون من 25 حرفاً أو أقل .
- (ج) حدد قنوات البيانات 2 و 5 لكل من ملفات البيانات العشوائية الممتلئة بالمتغيرات F\$ و G\$ على الترتيب .
- (د) أنشئ ملف البيانات المتسلسل TAPES احتفظ بالملف ثم اصدر قائمة بمحتوياته بعد الانتهاء من إدخال جميع البيانات .
- (هـ) سوف يكتب برنامج البيانات على ملف بيانات عشوائي ITEMS . بين كيف يمكن لملف بيانات خال يسمى ITEMS أن يعرف ويحفظ استمداداً لتشغيل البرنامج .
- (و) اقرأ برنامج البيانات من ملفات بيانات متتالية OLD 1 و OLD 2 ثم يكتب البيانات المعدلة على الملفات المتتالية NEW 1 و NEW 2 وفي وقت لاحق يمكن أن نطلب استخدام نفس البرنامج لقراءة ملف البيانات NEW 1 و NEW 2 . كيف يمكن تغيير الأسماء السابقة (NEW 1 و NEW 2) حتى يمكن استخدامها كملفات إدخال ؟

(ز) كيف يمكن إعادة كتابة البرنامج الموصوف في الجزء (و) حتى لا نحتاج إلى إعادة تسمية ملفات الإدخار قبل قراءتها؟ افترض أن ملفات الإدخال سوف تحدد بقنوات البيانات 1 و 2 وملفات الإدخار بالقنوات 3 و 4 .

(ح) يتكون كل سطر في ملف البيانات المتسلسل المحدد بقناة البيانات 5 من رقم السطر ، يتبعه قيم المتغيرات F\$ و X و Y و Z و G\$. افترض أننا نرغب في كتابة قيم Z و F\$ و G\$ على ملف بيانات متسلسل محدد بقناة البيانات 3 . بين كيف يمكن إنجاز ذلك .

(ط) إنسخ ملف البيانات المشوائي الرقعي المحدد بقناة البيانات 5 إلى الملف المحدد بقناة الاتصال 3 .

(ى) اقرأ كية صحيحة موجبة من الكونسول . حدد موضع المؤشر لقناة البيانات 6 إلى المكان المشار إليه بواسطة الكية المدخلة . اقرأ قيمة X من هذا الموضع ، ثم اكتب قيمة X في الموضع المناظر على قناة البيانات 2 .

(ك) حدد موضع مؤشر ملفات البيانات المشوائية المحددة بقنوات البيانات 1 و 4 حول التحكم إلى الجملة رقم 200 إذا كان كل من المؤشرين لها نفس القيمة (أى تشير إلى نفس الموضع بالترتيب) . وإلا اجعل قيمة مؤشر قناة البيانات 2 إلى القيمة الأكبر من القيمتين .

(ل) حدد المواضع الأخيرة من ملفات البيانات المشوائية التي تحددها قنوات البيانات 3,5 . حول التحكم إلى الجملة رقم 25 إذا كانت المواضع الأخيرة مختلفة عن بعضها .

مسائل للبرمجة

Programming Problems

٣٦-٨ عدل البرنامج المعطى في مثال ٨ - ٤ بحيث يمكن إدخال أسماء الملفات من الكونسول أثناء تنفيذ البرنامج . أيضاً ، ضمن اختياراً يسبب تخزين مجموعات من البيانات بترتيب متوسطات الفصل الدراسى وليس بترتيب أسماء الطلبة هجائياً .

٣٧-٨ عدل البرنامج المعطى في مثال ٨ - ١٠ حيث يمكن عمل أى من الاختبارات التالية :

(أ) ببساطة اطبع مستوى المخزون لأى رقم مخزون .

(ب) عرف « كتلة » من أرقام المخزون بقراءة أول وآخر رقم مخزون في الكتلة . اطبع مستوى المخزون لكل رقم مخزون في الكتلة .

(ج) اطبع مستوى المخزون لكل رقم مخزون في ملف البيانات .

(د) اطبع أرقام المخزون ومستويات المخزون المناظرة لكل البنود التي مستوى مخزونها أقل من مستوى مخزون محدد .

(هـ) اطبع أرقام المخزون ومستويات المخزون المناظرة لكل البنود التي مستوى مخزونها أكبر من مستوى مخزون محدد .

٣٨-٨ عدل البرنامج المعطى في مثال ٨ - ١٣ حتى يمكن القيام بأى من الاختبارات التالية :

(أ) إظهار قائمة كاملة لملف البيانات على الكونسول .

(ب) طباعة كل سلاسل الحروف مبتدئاً من حرف معين وذلك على الكونسول .

(ج) طباعة كل سلاسل الحروف التي تسبق سلسلة حرفية معينة .

(د) طباعة كل سلاسل الحروف الموجودة وراء نطاق سلسلة حرفية معينة .

٨ - ٣٩ أعد كتابة البرنامج الخاص بكل من المسائل التي تظهر قائمتها فيما يلي حيث يمكن أن تقرأ البيانات أو تكتب البيانات على ملف . (لاحظ : أن بعض المسائل سوف تتطلب قراءة البيانات من ملف بيانات وطباعة النتائج المحسوبة على الكونسول وهناك مسائل أخرى تقبل المدخلات من الكونسول ، وكتابة المخرجات على ملف البيانات . وفي حالات قليلة يمكن أن يكون المطلوب هو قراءة البيانات المدخلة من ملف بيانات ثم كتابة المخرجات على ملف بيانات آخر) .

- (أ) مثال ٤ - ١٦
 (ب) مثال ٤ - ١٨
 (ج) مسألة ٤ - ٤٨ (هـ)
 (د) مثال ٥ - ٥
 (هـ) مسألة ٥ - ٤٨
 (و) مسألة ٥ - ٤٩
 (ز) مسألة ٥ - ٥٠
 (ح) مسألة ٥ - ٥١
 (ط) مسألة ٥ - ٥٢
 (ي) مسألة ٥ - ٥٣
 (ك) مسألة ٥ - ٥٤
 (ل) مسألة ٥ - ٥٥
 (م) مسألة ٥ - ٥٦
 (ن) مسألة ٥ - ٥٧ (أ)
 (ش) مسألة ٥ - ٥٧ (ب)
 (س) مسألة ٥ - ٥٧ (ج)
 (ص) مسألة ٥ - ٥٧ (د)
 (ض) مسألة ٥ - ٥٧ (ح)
 (ط) مثال ٦ - ٢٦
 (ظ) مثال ٦ - ٢٨
 (ع) مسألة ٦ - ٤٨
 (غ) مسألة ٦ - ٥٢ (هـ)
 (ف) مسألة ٦ - ٥٢ (ي)

٨ - ٤٠ اكتب برنامج بييسك كاملا ينشأ وينتفع من ملف بيانات متسلسل يحتوي على أسماء وعناوين وأرقام تليفونات . ضمن احتياط لكل من الخصائص التالية :

- (أ) إضافة سجل جديد للملف (أى اسم جديد وعنوان ورقم تليفون) .
 (ب) أوجد سجلاً معيناً ثم اعرضه على النهاية الطرفية وذلك مبنى على أساس التعرف على المعلومات التي تدخلها من الكونسول (مثال ، إسم أو عنوان أو رقم تليفون) .
 (ج) احذف سجلاً معيناً ، مبنى على أساس التعرف على المعلومات التي تدخل من الكونسول .
 (د) رتب السجلات أبجدياً ، وذلك مبنى على أساس آخر اسم في كل سجل .
 (هـ) أعط قائمة (أى اطبع) الملف بالكامل .
 (و) أوقف الحسابات .

٨ - ٤١ كرر المسألة ٨ - ٤٠ مستخدماً ملف بيانات عشوائياً . ثم قارن نسخة ملف البيانات المتسلسل من وجهة نظر سهولة البرمجة وسرعة التنفيذ .

الجزء الثالث : بيسك الحاسب الدقيق

الفصل ٩

التحسينات على البيسك

Enhancements to BASIC

من المفترض أن كل الحاسبات الدقيقة تدعم البيسك كلغة برمجة ابتدائية . وتحتوى بعض الحاسبات الدقيقة على مفسر البيسك داخل ذاكرة للقراءة فقط (ROM) بحيث تصبح اللغة في الحقيقة جزءاً من دائرة الحاسب الداخلية . وتقرأ أجهزة أخرى مفسر البيسك ليدخل ذاكرة الحاسب من جهاز تخزين كبير (مثل قرص مرن) عندما نحتاجها . وفي كل من الحالتين ، فإن البيسك يطوع نفسه طبيعياً لبيفة الحاسب الدقيق . وفي الحقيقة فإن إتاحة البيسك أصبح من العوامل المعنوية جداً في التطوير التجارى لسوق الحاسب الدقيق .

ومعظم الحاسبات الدقيقة تدعم نسخ بيسك معقدة وتتضمن تحسينات كثيرة غير موجودة في النسخ التقليدية من اللغة . هذا بالإضافة إلى أن هذه الصور المعقدة من البيسك في معظمها هي أنواع مختلفة من ميكروسوفت بيسك (أعدتها مؤسسة Microsoft Corporation) . وعلى سبيل المثال فإن الحاسبات الدقيقة التي قامت بتسويقها شركات TEXAS INSTRUMENTS, ZENITH AND IBM AT& T, APPLE-RADIO (SHACK) من بين مجموعة أخرى كلها تتضمن بعض الاختلافات في ميكروسوفت بيسك . ولهذا فإن المادة التي سنعرضها في الفصول الأربعة التالية من هذا الكتاب ستكون مبنية على أساس الخصائص الموجودة في ميكروسوفت بيسك (على الأخص نسخة ميكروسوفت بيسك على حاسبات IBM الشخصية الشائعة) .

وتهدف المادة المعروضة في هذا المرجع إلى تمثيل هذه الخصائص المتاحة على حساب دقيق نموذجي . ومع كل فعل القارئ أن يفهم أن كل هذه الخصائص قد لا تكون منفذة على كل حاسب . هذا بالإضافة إلى أن هذه الخصائص المتاحة عموماً قد يتم تنفيذها بشكل مختلف نوعاً ما على كل حاسب . وعلى ذلك ، فإن هذه المادة لا بد من اعتبارها كنظرة عامة على بيسك الحاسب الدقيق بدلاً من اعتبارها وثيقة مرجعية دقيقة . وعلى القارئ إذا ما أراد معلومات خاصة حول نسخة معينة من اللغة أن يرجع إلى الكتيب المرجعي المناسب للمبرمج . وسنجد في ملحق (د) من هذا الكتاب تلخيصاً لأكثر الخصائص الشائعة لميكروسوفت بيسك .

٩-١ توسعات أولية في اللغة ELEMENTARY LANGUAGE EXTENSIONS

معظم النسخ المختلفة لبيسك الحاسب الدقيق تسمح عموماً بمدى أوسع في تعريف واستخدام المتغيرات والمعاملات عن زميلاتها التقليدية . بعض التوسعات الأكثر شيوعاً تم مناقشتها أدناه .

أسماء متغيرات أكبر Larger Variable Names

تسمح معظم نسخ بيسك الحاسب الدقيق بأسماء المتغيرات الأطول من حرفين . وفي الحقيقة ، فإن بعض نسخ اللغة لا تضع أى قيود على أقصى عدد مسموح به من الحروف . وهذه خاصية ملائمة جداً ، حيث أنها تسمح لأسماء المتغيرات بأن تناظر البنود التي تمثلها . ومع كل ، فعادة ما تكون الحروف العديدة الأولى معنوية في التعرف على المتغير . ويمكن أن يتراوح العدد الشائع للحروف المعنوية من 2 إلى 40 حرفاً . وقد يصل عدد الحروف لأسماء المتغيرات على حاسب IBM الشخصي إلى 40 حرفاً ، وكلها معنوية .

مثال ٩ - ١

فيما يلي عدة أسماء متغيرات :

AREA	NAME
SIZE	ADDRESS
TABLE	PATIENT
VECTOR	PAYROLL
VELOCITY	TAX

سوف نتعرف كثير من نسخ البيسك للحاسب الدقيق على كل من هذه الأسماء كمتغيرات منفصلة . ومع كل فإذا كان الحرفان الأولان فقط معنوين فإن المتغيرات TAX, TABLE, PAYROLL, PATIENT, VELOCITY, VECTOR لا يمكن تمييز بعضها عن بعض .

ويجب تجنب استخدام الكلمات الدالة في البيسك كأسماء متغيرات مثل (PRINT و DATA و NEXT الخ) لأنها سوف تسبب تشويشاً للمترجم ، وغالباً ماينتج عنها رسالة خطأ . وعموماً ، فيجب على المبرمج أن يحرص على عدم استخدام أى من الكلمات الدالة الأقل شيوعاً كأسماء متغيرات (مثل NAME, CHAIN, SWAP الخ) .

Multiple Numeric Data Types الأنواع المتعددة للبيانات الرقمية

تميز معظم نسخ البيسك للحاسب الدقيق بين الكميات الصحيحة والكميات الحقيقية والكميات المزدوجة الدقة .

والكميات الصحيحة هي الأرقام الصحيحة الكاملة سواء أكانت موجبة أم سالبة والتي تقع في المدى ما بين 32,768 — و 32,767 ومثل هذه الكميات لا تتعرض لأخطاء التقريب . من ثم فهي مفيدة كمعادلات أو أدلة لعناصر المجموعات المتراسة أو كتقنيات جارية (مؤشرات) في الحلقات التكرارية FOR-TO أو جمل ON GO TO و .. الخ . يكون من المستحب أيضاً استخدام الكميات الصحيحة عند القيام باختبارات منطقية معينة باستخدام جملة IF-THEN .

والكميات الحقيقية هي قيم رقمية تمت مناقشتها في الفصل الثاني . والكمية الحقيقية يمكن أن تتضمن أولاً تتضمن مكونات كسرية (عشرية) و/ أو أس ، وقد يكون مقدار الكمية الحقيقية صغيراً أى 2.9E-39 ، وكبيراً إلى 1.7E + 38 (والقيمة 0.0 هي أيضاً قيمة حقيقية مقبولة) .

وفي داخل ذاكرة الحاسب يتم تمثيل الكميات الحقيقية والكميات الصحيحة بطريقة مختلفة حتى إذا لم يكن هناك مركبة كسرية : وتكون الكميات الحقيقية غير دقيقة (تكون معرضة لأخطاء التقريب) بينما تكون الكميات الصحيحة مضبوطة .

والكميات المزدوجة الدقة هي أساساً كميات حقيقية (مفردة الدقة) ولكن لها عدد أكبر من الأرقام المنوية . والكمية المضاعفة الدقة النموذجية تمثل بدلالة 16 رقماً منوياً ، بينما الكمية الحقيقية سوف تحتوى فقط على 6 أو 7 أرقام منوية . وأيضاً يستخدم الحرف D لتمثيل الأس بدلا من الحرف E .

وتتعارف العديد من نسخ بييسك الحاسبات الدقيقة على أنواع أخرى من البيانات الرقمية مثل الثنائي (الأساس 2) والسادس عشر (الأساس 16) ، ومع كل فاستخدام أنواع البيانات هذه يخرج عن نطاق مناقشتنا الحالية .

مثال ٢ - ٩

عدة أنواع مختلفة من الأرقام مبينة فيما يلي :

النوع	الرقم
صحيحة أو حقيقية	16458
حقيق	36.55
حقيق	-0.666667E-3
مزدوج الدقة	-0.6666666666666667D-3

عندما نقوم بتمثيل رقم بمتغير رقمي فلا بد أن يكون الرقم والمتغير من نفس النوع وبذلك فإن الكميات الصحيحة والحقيقية والمزدوجة الدقة لا بد أن تمثل بمتغيرات صحيحة وحقيقية مزدوجة الدقة على الترتيب . وعموماً فإن الأنواع المختلفة من المتغيرات تعرف بواسطة آخر حرف من اسم المتغير . وكنموذج فإن المتغير الصحيح ينتهي دائماً بعلامة (%) ، وينتهي المتغير الحقيقي بعلامة التمجيد (!) ، والمتغير المزدوج الدقة ينتهي بعلامة الجنيه (£) . (تذكر أن المتغير الحرفي ينتهي دائماً بعلامة الدولار (\$)) . إن لم ينته اسم المتغير بأى حرف مميز ، فسوف يترجم على أنه متغير حقيق .

ويطلق على هذه النهايات الخاصة أحياناً ملاحق . والأنواع العديدة من الملاحق يمكن تلخيصها فيما يلي :

الملاحق	المتغير
%	صحيح
! (أو بدون ملحق)	حقيق (أو دقة مفردة)
#	متضاعف الدقة
\$	حرفي

مثال ٣ - ٩

عدة متغيرات بيسك وأنواع البيانات المناظرة لها مبينة فيما يلي :

نوع البيانات	المتغير
صحيحة	COUNT%
حرفية	NAME\$
حقيقية (مفردة الدقة)	PAY
حقيقية (مفردة الدقة)	TAX !
متضاعف الدقة	ERROR #

وفي حالة نفس الاسم وبملاحق مختلفة سوف يترجم على أنه متغير مختلف .

مثال ٤ - ٩

نفرض أن المتغيرات

A و A\$ و A% و A #

كلها تظهر في نفس برنامج البيسك . وسوف تترجم كل منها كمتغير مستقل ومنفصل حيث أن لها ملاحق مختلفة ، ومن ثم فإنها تمثل أنواع بيانات مختلفة .

ويجب أن يعرف المبرمج أن متطلبات الذاكرة لأنواع البيانات المختلفة ليست متماثلة ، فتنطلب الكميات الصحيحة أقل مساحة من الذاكرة 2 bytes ، بينما تنطلب الكميات المزدوجة الدقة أكبر مساحة من الذاكرة 8 bytes . وأيضاً ، فإن أسماء المتغيرات الطويلة تنطلب ذاكرة أكبر من التي تنطلبها أسماء المتغيرات القصيرة تماماً كما تنطلب السلاسل الحروف الطويلة ذاكرة أكبر من السلاسل الحروف القصيرة . ويجب أن تؤخذ هذه العوامل في الاعتبار عند استخدام الحاسبات الدقيقة حيث أن كمية الذاكرة المتاحة يمكن أن تكون قليلة .

ويجب تجميع مثل هذه الاعتبارات لوقت التنفيذ . وكقاعدة ، فإن البرامج التي تستخدم متغيرات صحيحة يتم تشغيلها أسرع من البرامج التي تحتوي على متغيرات حقيقية أو مزدوجة الدقة . وبذلك فإن البرنامج الذي يتضمن حسابات رقمية بكفاءة كبيرة يجب أن ينتفع بالمتغيرات الصحيحة كلما كان ذلك عملياً .

العمليات الحسابية المختلطة Mixed Arithmetic Operations

عند تنفيذ عمليات حسابية بها بيانات رقمية مختلفة النوع يتم التعبير عن نتائجها بأعلى مستوى من الدقة . وعلى ذلك فالعمليات الحسابية التي تشتمل على بيانات صحيحة وحقيقية تكون نتائجها حقيقية ، وبالمثل فالعمليات الحسابية التي تستخدم إما بيانات صحيحة ومزدوجة الدقة أو بيانات حقيقية ومزدوجة الدقة سوف تكون نتائجها مزدوجة الدقة .

مثال ٩ - ٥

في كل من التعميرات التالية نفرض أن $I\% = 4$, $R! = -0.2$ and $D\# = 0.16666666D-4$

التعبير	القيمة
$R! + 5$	4.8 حقيقية
$I\% * R!$	-0.8 حقيقية
$3 * I\% * D\#$	0.199999992D-3 مزدوجة الدقة
$(1 + R!) * D\#$	0.133333328D-5 مزدوجة الدقة

والآن ادرس جملة التحديد الرقمية (أى جملة LET). حيث المتغير الموجود على اليسار والقيمة الموجودة على اليمين من أنواع مختلفة . سوف تحول القيمة الموجودة في اليمين أتوماتيكياً لنفس نوع المتغير الموجود في اليسار . لاحظ أن ذلك سوف يتسبب في تقريب القيمة الكسرية الموجودة في اليمين إذا ظهر متغير صحيح في اليسار (بعض نسخ البيسك سوف تقوم بالتردد بدلاً من التقريب للقيمة الكسرية) .

مثال ٩ - ٦

في كل من جمل التحديد التالية ، نفرض أن $I\% = 4$ و $R! = -0.2$ و $D\# = 1.6666666 D-4$.

جملة التحديد	النتيجة
10 LET $N\% = 3 * R!$	$N\% = -1$
20 LET $FRACT = 1/3$	$FRACT = 0.3333333$
30 LET $FRACT\# = 1/3$	$FRACT\# = 0.3333333333333333$
40 LET $ANS\% = I\% \uparrow 2/3$	$ANS\% = 5$
50 LET $RATIO = (I\% - R!)/D\#$	$RATIO = 2.52E+5$

(لاحظ التقريب الذي نتج عند تنفيذ الجملة رقم 10 . كان البتر سيؤدي إلى $N\% = 0$ بدلاً من $N\% = -1$) .

تذكر أن القيمة الرقمية لا يمكن أن تستخدم لتحديد متغير حرفي ، وبالعكس . وتذكر أيضاً أن الكلمة المرشدة LET اختيارية في العديد من نسخ البيسك للحاسبات الدقيقة .

معاملات إضافية Additional Operators

تتضمن بعض نسخ البيسك معاملين حسابيين إضافيين : وهما القسمة الصحيحة (\) ، والكية المتبقية الصحيحة (MOD) . في القسمة الصحيحة يقرب كل من الرقين المعطيين أولاً لرقم صحيح ، ثم تنفذ عملية القسمة ، ثم يترك خارج القسمة بعد ذلك . تمدنا عملية إيجاد الكية المتبقية الصحيحة بالكية المتبقية بعد تنفيذ عملية القسمة الصحيحة .

مثال ٩ - ٧

تمثل فيما يلي عدة عمليات للقسمة الصحيحة وإيجاد الباقي الصحيح :

$$\begin{array}{ll} 13 \setminus 5 = 2 & 13 \text{ MOD } 5 = 3 \\ 8.6 \setminus 2.7 = 3 & 8.6 \text{ MOD } 2.7 = 0 \\ 8.3 \setminus 2.7 = 2 & 8.3 \text{ MOD } 2.7 = 2 \\ 8.3 \setminus 2.2 = 4 & 8.3 \text{ MOD } 2.2 = 0 \end{array}$$

بعض النسخ الأكثر بساطة من بيسك الحاسبات الدقيقة تتعرف فقط على البيانات الرقية من النوع الصحيح . في مثل هذه الحالات فإن معامل القسمة العادي (\) سوف يعنى عملية القسمة الصحيحة (مع ترك خارج القسمة) .

وتتضمن بعض نسخ البيسك للحاسبات الدقيقة المعاملات المنطقية مثل AND و OR و NOT . أو معاملين (AND و OR) تسمح بدمج شرطين أو أكثر من شروط « صحيحة خاطئة » . ويستخدم المعامل الثالث (NOT) لنفى شرط « صحيح — خطأ » (أى يغير « الصحيح » إلى « خطأ » أو « الخطأ » إلى « صحيح ») تسمح هذه العمليات بتضمين شروط أكثر تعقيداً في جملة IF-THEN .

مثال ٩ - ٨

مبين فيما يلي عدة جمل IF-THEN تستخدم الشروط المنطقية المقيدة .

```
10 IF (X<10) AND (Y>0) THEN 90
20 PRINT X,Y
```

(أ)

سوف تنفذ الجملة رقم 90 مباشرة إذا كانت قيمة X أقل من 10 وقيمة Y أكبر من الصفر ، وإلا فسوف تنفذ الجملة رقم 20 مباشرة .

```
50 IF (COUNT>99) OR (N$ = "END") THEN 175
60 GO TO 30
```

(ب)

سوف تنفذ الجملة رقم 175 مباشرة إذا تعدت قيمة COUNT الرقم 99 أو إذا كانت قيمة N\$ هى "END" (أو كليهما) . أما إذا كان كلا الشرطين غير صحيح فسوف تنفذ الجملة رقم 60 مباشرة .

```
75 IF NOT ((X<10) AND (Y>0)) THEN 200
80 PRINT X,Y
```

(ج)

هذا المثال عكس المثال (أ) . في الحالة الراهنة سوف تنفذ الجملة رقم 200 إذا كان الشرط :

(X<10) AND (Y>0)

غير صحيح ، أى إذا كانت قيمة X أكبر أو تساوى 10 أو إذا كانت قيمة Y أقل من أو تساوى 0 ، وإلا فسوف تنفذ الجملة رقم 50 مباشرة . لاحظ أن جملة IF كان يمكن كتابتها أيضاً :

```
75 IF (X >= 10) OR (Y <= 0) THEN 200
```

وكتطبيق عمل فإن معامل NOT يلتق استخداماً أقل نسبياً في برامج البيسك الحقيقية .

وتحتوى بعض نسخ البيسك للحسابات الدقيقة على معاملات إضافية منطقية XOR (Exclusive OR) ، EQV (Equivalent) ، IMB (Implies) وباختصار عندما تستخدم لتوصيل عاملين منطقيين، فإن XOR تنتج في حالة. صحيحة إذا كان أحد العوامل صحيح والآخر غير صحيح ، وتنتج عن EQV شرط إذا كان كل من العاملين له نفس القيمة المنطقية (إما صحيحين أو غير صحيحين) أو ينتج عن IMB حالة صحيحة إذا كان كل من العاملين أو إذا كان العامل الأول غير صحيح (بصرف النظر عن قيمة العامل الثانى) . وحيث أن هذه العوامل المنطقية الثلاثة لا تستخدم إلا قليلاً فإننا ستجنب مناقشتها في هذا النص .

والتدرج الهرمى النموذجى الكامل للمعاملات الحسابية والمنطقية ومعاملات الترابط هو :

١ - الرفع للأس	↑
٢ - النفى (أى ، أن يسبق المتغير علامة الطرح)	
٣ - الضرب والقسمة	*/
٤ - القسمة الصحيحة	\
٥ - الباقى الصحيح .	MOD
٦ - الجمع والطرح	+ -
٧ - عمليات الترابط	= <> <= < >= >
٨ - NOT المنطقية	NOT
٩ - AND المنطقية	AND
١٠ - OR المنطقية	OR
١١ - XOR المنطقية	XOR
١٢ - EQV المنطقية	EQV
١٣ - IMP المنطقية	IMP

ويمكن أن يتغير التسلسل الهرمى من نسخة بيسك لأخرى . وبداخل مجموعة التسلسل الهرمى تنفذ العمليات من اليسار إلى اليمين .

عدة نسخ معدلة من البيسك يمكن أن تتضمن أيضاً معامل الوصل وهو يستخدم في دمج السلاسل الحرفية . وغالباً مايمثل هذا المعامل بواسطة علامة الجمع (+) ، بالرغم من أن علامة & أو الفصلة (,) يمكن أن تستخدم لنفس الغرض في نسخ معينة من اللغة .

مثال ٩ - ٩

نفرض أن المتغيرات الحرفية A \$ و B \$ تمثل بالسلاسل الحرفية "MICRO" و "COMPUTER" على الترتيب . وبذلك فإن الجملة

10 PRINT A \$ + B \$

سوف تسبب طباعة سلسلة حرفية (واحدة) وهى :

MICROCOMPUTER

أن تطبع . (نفرض أن رمز الوصل هو علامة الجمع +) .

وبالمثل .. فإن الجملة

```
20 PRINT "SEVEN" + "TEEN"
```

سوف يترتب عليها ظهور السلسلة الحرفية الواحدة :

SEVENTEEN

جمل متعددة في السطر Multiple Statements per Line

تسمح معظم نسخ بيسك المعدلة بظهور جمل متعددة في نفس السطر . وعادة ما تكون علامة الوقف الاستدراكي (;) هي التي تستخدم في فصل إحدى هذه الجمل عن الأخرى .

مثال ٩ - ١٠

عدة سطور تحتوى على جمل متعددة مبنية فيما يلي :

```
10 LET A=0.25 : B=0.5 : C=-0.125
20 PRINT "X="; : INPUT X
30 FOR I%=1 TO N% : READ A(I%) : NEXT I%
```

واستخدام هذه الخاصية يلائم غرضين : أولاً أنه يسمح لمجموعة متتالية من الجمل المترابطة أن تظهر في صورة مجموعة منطقية ، وثانياً ، أنه يساهم في استخدام الذاكرة بطريقة أكثر فاعلية . وعادة استخدام السطور العديدة الجمل في برامج يمكن أن تترتب عليه صعوبة نسبية في قراءة وتصحيح البرنامج .

مثال ٩ - ١١ : توليد أرقام فيبوناتشى والبحث عن الأرقام الأولية على الحاسب الدقيق .

Generating Fibonacci Numbers and Searching for Primes on a Microcomputer

رأينا في المثال ٤ - ١٨ برنامج بيسك كاملاً يولد مجموعة متتالية من أرقام فيبوناتشى والتعرف على الأرقام الأولية من بينها . وبين شكل ٩ - ١ برنامج بيسك آخر مكتوباً بميكروسوفت بيسك الذى يسمح بالقيام بنفس الحسابات على حاسب IBM الشخصى . يستخدم هذا البرنامج توسعات اللغة العديدة التى تم عرضها سابقاً في هذا الفصل . وعلى الأخص ، فإننا نرى استخدام أسماء المتغيرات الطويلة ، والمتغيرات من النوع الصحيح ، والجمل المتعددة في عدة سطور ، وأيضاً استخدام المنطق للتعرف على الأرقام الأولية ، حيث أنه من الممكن استخدام عملية باقى القسمة الصحيحة (MOD) . (قارن السطور 110 إلى 120 في البرنامج الحالى مع السطور 140 إلى 160 في البرنامج الأصل) .

```
10 REM ***** GENERATION OF FIBONACCI NUMBERS AND SEARCH FOR PRIMES *****
20 PRINT "Generation of Fibonacci Numbers and Search for Primes";PRINT
30 PRINT "How many Fibonacci numbers";:INPUT NX;PRINT
40 F1%=1:F2%=1
50 PRINT "I=";1,"F=";F1%;" (Prime)"
60 PRINT "I=";2,"F=";F2%;" (Prime)"
70 FOR INDEX%=3 TO NX
80   FX=F1%+F2%;ROOT=SQR(FX);MAXX=ROOT
90   PRINT "I=";INDEX%,"F=";FX;
100  FOR DENOMX=2 TO MAXX
110    REMAINDERX=FX MOD DENOMX
120    IF REMAINDERX=0 THEN 150
130  NEXT DENOMX
140  PRINT " (Prime)";
150  PRINT
160  F2X=F1X:F1X=FX
170 NEXT INDEX%
180 END
```

شكل ٩ - ١

وينتج تنفيذ هذا البرنامج نفس المخرجات المبينة في شكل ١٠-٤ من المثال ٤-١٨ . وشكل ٩-٢ يبين نموذجاً لمجموعة مخرجات يولدها هذا البرنامج (رغم أن المخرجات عادة يتم عرضها على شاشة TV بدلاً من طباعتها على نسخة ورقية على النهاية الطرفية . واستجابات المستخدم موضوع تحتها خط .

ويجب أن نذكر هنا ، أنه بالنسبة للحاسب الدقيق ذي الأرقام الثنائية تكون أكبر قيمة يمكن إعطاؤها للمتغير N% (في السطور رقم 30) هو 23 ، حيث أن أي قيمة أكبر من 23 سوف يترتب عليها رقم من أرقام فيبوناتشي يتعدى 32,767 وأكبر رقم صحيح مسموح به على معظم الحاسبات الدقيقة. ويمكن لهذا القيد أن يحد من الاستخدام العمل لهذا البرنامج على حاسب دقيق ذي ثمانية أرقام ثنائية. لاحظ أن السلسلة المبينة في السطور 20, 30, 50, 60, 140 تحتوي على كل من الحروف الكبيرة والصغيرة والتي تفرق بينها معظم الحاسبات الدقيقة في حالة السلاسل فقط .

Generation of Fibonacci Numbers and Search for Primes

How many Fibonacci numbers? 23

I= 1	F= 1 (Prime)
I= 2	F= 1 (Prime)
I= 3	F= 2 (Prime)
I= 4	F= 3 (Prime)
I= 5	F= 5 (Prime)
I= 6	F= 8
I= 7	F= 13 (Prime)
I= 8	F= 21
I= 9	F= 34
I= 10	F= 55
I= 11	F= 89 (Prime)
I= 12	F= 144
I= 13	F= 233 (Prime)
I= 14	F= 377
I= 15	F= 610
I= 16	F= 987
I= 17	F= 1597 (Prime)
I= 18	F= 2584
I= 19	F= 4181
I= 20	F= 6765
I= 21	F= 10946
I= 22	F= 17711
I= 23	F= 28657 (Prime)

شكل ٩-٢

٩ - ٢ جمل إضافية ADDITIONAL STATEMENTS

تتضمن معظم نسخ البيسك للحاسبات الدقيقة عدداً من الجمل المناسبة غير الملائمة وغير المتاحة على نسخ اللغة التقليدية ، العديد من الجمل الإضافية الأكثر شيوعاً سوف تناقش أدناه .

DEFINT, DEF\$NG, DEFDBL, and DEFSTR

يكون من المناسب في بعض الحالات تعريف عدة متغيرات مختلفة لأنواع بيانات معينة . ويمكن إنجاز ذلك مع جمل DEFINT و DEF\$NG و DEFDBL و DEFSTR (للصحيح ومنفرد الدقة ومزدوج الدقة وبيانات السلاسل الحرفية على الترتيب) بشرط أن كل أسماء المتغيرات لنوع معين تبدأ بنفس الحرف الأول . ومن الممكن أيضاً تعريف مدى للحروف الأولى ، كما هو موضح أدناه .

مثال ٩ - ١٢

برنامج ببسك يحتوى على الجمل التالية :

```
10 DEFINT I
20 DEFDBL X-Z
30 DEFSTR A-C,N
```

تنص الجملة رقم 10 على أن كل المتغيرات المبتدئة بالحرف I سوف تكون متغيرات صحيحة . ويترتب على الجملة رقم 20 أن كل المتغيرات المبتدئة بالحروف X و Y و Z تكون متغيرات مزدوجة الدقة ، وتعرف الجملة رقم 30 أن كل المتغيرات المبتدئة بالحروف A و B و C و N على أنها متغيرات حرفية . (لاحظ أنه لايلزم استخدام الملاحق لأى من هذه المتغيرات)

ويجب أن يفهم أن أسماء المتغيرات التى تتضمن ملحقاً لها أسبقية على أسماء المتغيرات التى تعرف بواسطة جملة نوع - DEF .

IF - THEN

تتضمن معظم نسخ البيسك للحاسبات الدقيقة شكلاً موسعاً من جملة IF-THEN حيث يمكن أن تظهر جملة مستقلة أو أكثر بعد الكلمة الدالة THEN على شرط أن تظهر كلها على نفس السطر . سوف تنفذ هذه الجمل إذا استوفى الشرط المعطى ، وإلا فسوف تنفذ الجملة المبتدئة على السطر التالى مباشرة .

مثال ٩ - ١٣

يحتوى برنامج ببسك على الجمل التالية :

```
50 IF ERROR>0.001 THEN PRINT ERROR
60 LET X = X1
```

سوف تتسبب جملة IF-THEN فى طباعة قيمة ERROR إذا تعدت الكمية 0.001 ، وإلا فسوف يتحول التحكم مباشرة إلى السطر رقم 60 .

مثال ٩ - ١٤

يحتوى برنامج ببسك على الجملة التالية :

```
200 IF FLAG = 1 THEN A = 10 : C$ = "BLUE" : GO TO 35
210 PRINT A
```

إذا كانت FLAG لها القيمة 1 ، فإن المتغيرات A و G سوف تملى القيم 10 و BLUE على الترتيب ، وبعد ذلك يتحول التحكم إلى السطر رقم 35 وإلا ، فسوف ينفذ السطر رقم 210 بعد ذلك (لاحظ أن الكلمة الدالة THEN تتبعها ثلاث جمل منفصلة) .

IF-THEN-ELSE

تسمح بعض نسخ ببسك للحاسبات الدقيقة بتضمين عبارة ELSE فى جملة IF-THEN ، وبذلك فإن الجملة (أو الجمل) التى تتبع الكلمة الدالة THEN سوف تنفذ إذا توفر الشرط المعطى، وإلا فإن الجملة (أو الجمل) التى تتبع ELSE هى التى سوف تنفذ .

مثال ٩ - ١٥

يحتوى برنامج ببسك على جملة IF-THEN-ELSE التالية :

```
80 IF Z < 0 THEN 50 ELSE 120
```

إذا كانت قيمة Z أقل من الصفر ، فإن الجملة الموجودة في السطر رقم 50 هي التي تنفذ مباشرة ، وإلا فسوف تنفذ الجملة الموجودة في السطر رقم 120 بعد ذلك .

مثال ٩ - ١٦

والآن إدرس جملة بيسك التالية :

80 IF DIFF<0.001 THEN PRINT ANS: GO TO 300: ELSE X=X1: GO TO 35

سوف تطبع قيمة ANS ثم يتحول التحكم إلى السطر رقم 300 إذا كانت قيمة DIFF "أقل من 0.001 ، وإلا فسوف نحدد قيمة X بالقيمة الحالية للمتغير X1 ثم يتحول التحكم إلى السطر رقم 35 .

وعادة ، يمكن أن تكون جملة IF-THEN-ELSE متداخلة مع جملة IF-THEN-ELSE أخرى ، بالرغم من أن المنطق يمكن أن يكون غامضاً ويتطلب مهارة فائقة ويمكن أن تظهر النتائج مختلفة تماماً عما يمكن أن يتوقعها المبرمج . ويجب أن يرجع القارئ إلى المرجع الخاص به من أجل معلومات أكثر عن جملة IF-THEN-ELSE المتداخلة .

وأخيراً .. يجب أن نذكر أن استخدام جملة IF-THEN-ELSE الممتدة غالباً ما تحسن وضوح المنطق في برنامج بيسك . ويتم إنجاز ذلك بكتابة التفرعات المشروطة بطريقة مرتبة ومتتالية ، وبذلك ننقص من عدد جملة GO TO غالباً ما يطلق على تنظيم البرامج الطريقة البرمجة الهيكلية .

ON-GOSUB

جملة ON-GOSUB متاحة في العديد من نسخ بيسك الحاسبات الدقيقة . وهذه الجملة تشبه جملة ON-GO TO ما عدا أن التحكم يحول إلى برنامج فرعي بدلاً من جزء آخر من البرنامج الأصلي . وعند الانتهاء من البرنامج الفرعي ، فسوف يتحول التحكم مرة ثانية إلى الجملة التي تلي جملة ON-GOSUB مباشرة .

مثال ٩ - ١٧

يحتوي برنامج بيسك على الجمل التالية :

100 ON FLAG GOSUB 800, 1000, 1200
110 PRINT NAME\$

يترتب على جملة ON-GOSUB تحويل التحكم إلى أحد ثلاثة برامج فرعية مختلفة ، معتمداً في ذلك على القيمة المعطاة للمتغير FLAG. فإذا كان FLAG = 1 ، فسوف يتحول التحكم إلى البرنامج الفرعي الذي يبدأ عند السطر رقم 800 . وبالمثل فإن التحكم سوف يتحول إلى البرنامج الفرعي البادئ في السطر رقم 1000 إذا كان FLAG = 2 وإلى البرنامج الفرعي البادئ في السطر رقم 1200 إذا كان FLAG = 3. لاحظ أن الجملة رقم 110 التي تشمل على (PRINT NAME\$) سوف تنفذ بعد البرنامج دون اعتبار لأي برنامج فرعي تم اختياره .

ON ERROR GO TO

تعرف معظم نسخ بيسك للحاسبات الدقيقة على عدد من أنواع مختلفة من الأخطاء التي يمكن أن تحدث أثناء تفسير البرنامج أو تشغيله . وجملة ON ERROR GO TO تستخدم لأجراء تصحيح الخطأ بمجرد اكتشافه ، وعلى الأخص فإن جملة ON ERROR GO TO تقوم بتحويل التحكم أوتوماتيكياً لجزء بعيد في البرنامج عند اكتشاف الخطأ على شرط أن يحدث اكتشاف الخطأ بعد جملة ON ERROR GO TO ويعرف ذلك بتصيد الخطأ . وتسمح بتضمين رسائل خطأ أو برامج فرعية لتصحيح الخطأ بداخل البرنامج ، وعادة ما تستخدم جملة RESUME مقترنة بجملة ON ERROR GO TO (انظر ما يلي) .

RESUME

تُستخدم جملة **RESUME** للإشارة إلى المكان الذي يجب أن نكمل من عنده التنفيذ بعد اكتشاف خطأ ما وبعد تنفيذ البرنامج لتصحيح الخطأ.

مثال ٩ - ١٨

مبين فيما يلي جزء لتصحيح الأخطاء من برنامج بيسك :

```
10 ON ERROR GO TO 800
:
50 PRINT "ACCOUNT NUMBER: ";
60 INPUT ACCTNO
:
800 PRINT "INPUT ERROR—TRY AGAIN"
810 RESUME 50
```

والآن نفرض أننا أدخلنا بيانات إدخال غير صحيحة أثناء تشغيل البرنامج (أدخلنا - مثلاً - سلسلة حروف بدلاً من رقم في السطر رقم 60) . وذلك سوف يسبب تفرعاً إلى السطر رقم 800 ، وينتج عنه توليد رسالة خاطئة وسوف يرجع التحكم مرة أخرى للسطر رقم 50 لمحاولة أخرى لقراءة البيانات بطريقة صحيحة .

وستحدث بإفازة حول استخدام جملتي **ON ERROR GO TO** و **RESUME** في الفصل الحادى عشر [انظر قسم ١١ - ٣] .

WHILE and WEND

تتضمن نسخ عديدة من البيسك للحاسبات الدقيقة خاصية تكوين الحلقات التكرارية المشروطة حيث تنفذ مجموعة متتالية من الجمل مراراً ، مادام شرط معين يظل صحيحاً . تستخدم جمل **WHILE** و **WEND** لتعريف البداية والنهاية على الترتيب للحلقة التكرارية المشروطة . والشرط الذى يجب أن يستوفى (أى يبقى صحيحاً) يكون متضمناً فى جملة **WHILE** . وفى داخل الحلقة التكرارية لا بد من إيجاد وسيلة لتغيير الشرط وإلا فإن الحلقة ستستمر إلى ما لا نهاية .

مثال ٩ - ١٩

مبين فيما يلي حلقة تكرارية مشروطة وهى تجمع عناصر المصفوفة العددية **X** ، مادامت قيمة **x** أكبر من الصفر .

```
10 DIM X(100)
:
100 LET SUM = 0 : I = 1
110 WHILE X(I) > 0
120     SUM = SUM + X(I) : I = I + 1
130 WEND
```

تبدأ الحلقة التكرارية المشروطة فى هذا المثال بالسطر رقم 110 وتنتهى بالسطر رقم 130 . لاحظ استخدام الجمل المتعددة فى السطور 100 و 120 . لاحظ أيضاً أن الكلمة الدالة **LET** قد حذفت من السطر 120 . لا بد من ملاحظة أن هذا المثال سوف يولد خطأ إذا كانت كل عناصر الصف موجبة حيث أن الحلقة التكرارية سوف تستمر فى التنفيذ حتى يزيد دليل الصف (I) على الحد الأعلى (100) المحدد فى جملة **DIM** .

INPUT

في عديد من نسخ البيسك للحاسبات الدقيقة ، يمكن استخدام جملة **INPUT** لطباعة رسالة تلقين (أى سلسلة حروف) قبل إدخال بيانات الإدخال. ولعمل ذلك ، يجب أن تتبع الرسالة الكلمة الدالة **INPUT** ويجب أن تكون محصورة بين علامتى اقتباس . وغالباً ما يحتاج الفاصلة المنقوطة (؛) بعد الرسالة ، وذلك بفرض فصل سلسلة الحروف من قائمة متغيرات الإدخال .

مثال ٩ - ٢٠

10 INPUT "WHAT IS YOUR NAME"; N\$

إدرس الجملة :

وعند تنفيذ هذه الجملة تظهر الرسالة التالية على النهاية الطرفية :

WHAT IS YOUR NAME?

ورداً على ذلك (أى سلسلة الحروف التى تعطى للمتغير N\$) فسوف ندخل قيمة المتغير N\$ على نفس السطر وتلى مباشرة علامة الاستفهام . وبذلك ، إذا اختار المستخدم الإجابة SANTA CLAUS فتطبع هذه القيمة ويظهر السطر كاملاً كالتالى :

WHAT IS YOUR NAME? SANTA CLAUS

(لاحظ أن إجابة المستخدم قد وضع تحتها خط) .

وتتضمن بعض نسخ بيسك فرصة اختيارية لعدم ظهور علامة الاستفهام بعد رسالة التلقين . ويمكن إنجاز ذلك عادة بوضع الفاصلة (،) بدلاً من الفاصلة المنقوطة (؛) عند نهاية رسالة التلقين .

مثال ٩ - ٢١

الآتى اختلاف فى جملة **INPUT** عن مثال ٩ - ٢٠ . وفى هذا المثال ستحذف علامة الاستفهام التى تتبع جملة التلقين .

10 INPUT "PLEASE ENTER YOUR NAME: ",N\$

يؤدى تنفيذ هذه الجملة إلى العبارة التالية

PLEASE ENTER YOUR NAME:

للظهور على النهاية الطرفية ، فإذا استجاب المستخدم ثانية بإدخال اسم SANTA CLAUS فإن السطر كله سيظهر كالتالى :-

PLEASE ENTER YOUR NAME: SANTA CLAUS

(استجابة المستخدم تحتها خط) .

لاحظ أن عبارة التلقين قد تم تعديلها لتظهر كجملة بدلاً من سؤال حيث لا توجد علامة استفهام مرتبطة بهذا التلقين .

INKEY\$ and INPUT\$

تستخدم الدالتان (وليس الجملتان) **INKEY\$** و **INPUT\$** لإدخال حرف مفرد أو سلسلة حروف متعددة على الترتيب من لوحة المفاتيح ، وحتى يمكن استخدام هذه الدوال بطريقة سليمة فلا بد من تخصيص سلسلة الحروف التى أدخلت إلى متغير لسلسلة الحروف المناسبة . واستخدام هذه الدوال لا يولد علامة استفهام عندما تطلب بيانات مدخلة وذلك لا يماثل جملة **INPUT** . وسلسلة الحروف المدخلة تدخل

ببساطة من لوحة المفاتيح بدون الضغط على مفتاح الرجوع وسوف لا تعرض الحروف المدخلة على النهاية الطرفية .
وتستخدم أحياناً الدالتان INPUT\$, INKEY\$ لإيقاف تنفيذ البرنامج ، وعلى ذلك فإن تنفيذ البرنامج سيتوقف مؤقتاً عند مقابلة إحدى هاتين الدالتين حتى يتم إدخال سلسلة الحروف المطلوبة. وتحتاج INKEY\$ لسلسلة حروف فردية ، بينما تحتاج INPUT\$ لسلسلة حروف يتحدد طولها كجزء من الدالة المشار إليها . ويحتوى قسم ٩ — ٣ على مناقشة أكثر عمومية على الدوال في بيסק الحاسبات الدقيقة .

مثال ٩ — ٢٢

يحتوى برنامج بيסק للحاسبات الدقيقة على عدد من جمل PRINT يقصد بها إمداد مستخدمى البرنامج بمجموعة من التعليمات . وسوف تملأ هذه التعليمات شاشة TV كاملة (ومن ثم تختفى فور مسح الشاشة) . ومن أجل إعطاء المستخدم وقتاً كافياً لقراءة هذه التعليمات فإن جمل الطباعة تتبعها جملة لاستخدام دالة INKEY\$ ، ومن ثم فإن التعليمات ستبقى على الشاشة حتى يضغط المستخدم على مفتاح مسبباً إدخال سلسلة حروف مفردة إلى الحاسب .

مبين فيما يلى جزء من البرنامج .

```
10 PRINT TAB (34); "INSTRUCTIONS"
:
190 PRINT TAB(26); "(PRESS ANY KEY TO CONTINUE)"
200 A$=INKEY$ : IF A$="" THEN 200
```

لاحظ أن البرنامج سوف يستمر فى الدوران فى الحلقة خلال جملة 200 حتى يتم الضغط على أحد المفاتيح الذى يتسبب فى إدخال سلسلة الحروف الفردية المطلوبة والتي يتم تخصيصها للسلسلة .
وهنا طريقة أخرى لإنجاز نفس الشيء باستخدام دالة INPUT\$.

```
20 PRINT TAB (34); "INSTRUCTIONS"
:
190 PRINT TAB(26); "(PRESS ANY KEY TO CONTINUE)"
200 A$=INPUT$(1)
```

فى هذه الحالة سيصل البرنامج إلى حالة توقف حتى يتم إدخال سلسلة حروف فردية من لوحة المفاتيح (1 الذى يظهر بين القوسين بعد INPUT\$ يعنى أن السلسلة المدخلة تحتوى على حرف واحد) . وستتناول باستفاضة هذه الأنواع من طرق البرمجة فى الفصلين العاشر والحادى عشر .

PRINT USING

جملة PRINT USING متاحة فى عديد من نسخ البيسك للحاسبات الدقيقة وهى تسمح بطباعة المخرجات فى صيغة معينة ، وبذلك تصف شكل ومكان كل عنصر من البيانات . ويمكن صياغة كل من البيانات العددية والحرفية ، إلا أن هذه الخاصية عامة تكون أكثر فائدة فى البيانات الرقمية .

وتوجد عدة طرق مختلفة لصياغة بيانات الإخراج بواسطة جملة PRINT USING . وتتضمن كل هذه الطرق وضع صيغة سلسلة حروف بعد الكلمة الدالة PRINT USING مباشرة وقبل قائمة بنود المخرجات . ويجب أن تظهر الفاصلة المنقوطة (;) بين صيغة سلسلة الحروف وبين أول بند من المخرجات .

مثال ٩ - ٢٣

وإليك أكثر الصيغ شيوعاً لجملة PRINT USING

100 PRINT USING "###.###";A,B,C

في هذا المثال سلسلة الحروف المصاغة هي "###.###" حيث تصف حقلاً رقمياً يحتوي على علامة عشرية من خانتين فقط على جانبي العلامة العشرية. وسوف تقرب الكسور (الأرقام العشرية) التي تمتد إلى ما وراء نطاق الخانتين. والمكان الخالي في نهاية سلسلة الحروف المصاغة يجب أن يكون موجوداً وذلك ليفصل بين القيم المطبوعة لكل من A و B و C.

والآن نفرض أن قيم المتغيرات A و B و C هي 17.667 و 5.38 و 40 على الترتيب، فسوف يتم توليد سطر المخرجات التالي باستخدام جملة PRINT USING التي ظهرت أعلاه :

17.67 -5.38 40.00

مثال ٩ - ٢٤

صيغة أخرى لجملة PRINT USING مبينة فيما يلي :

200 PRINT USING "###.###^"; VALUE

لاحظ الأسهم الأربعة الرأسية عند نهاية سلسلة الحروف المصاغة وهي تصف الترميز الأسّي. وبذلك إذا كانت قيمة المتغير VALUE بمثلة بالرقم 856.07 فإن جملة PRINT USING السابقة سوف تولد القيمة :

8.561E+02

وبالمثل إذا كانت قيمة المتغير VALUE هي الرقم 8560.7 - فإن جملة PRINT USING سوف تولد :

-8.561E+02

مثال ٩ - ٢٥

هذا المثال يوضح كيف يمكن إضافة الفاصلات (,) في أي رقم من المخرجات.

30 PRINT USING "#####.##"; COST#

الفاصلة التي تسبق العلامة العشرية في سلسلة الحروف المصاغة سوف تسبب وضع الفاصله (,) في جزء الرقم الصحيح لكل كمية من المخرجات، وبذلك إذا كان المتغير المزدوج النقة ~~COST~~ (قيمته هي 15673088.209) فإن الجملة السابقة سوف تولد المخرج التالي :

15,673,088.21

لاحظ أن الجزء الصحيح من الرقم قد تم تقسيمه إلى مجموعات من ثلاث خانات مبتدئاً من العلامة العشرية ومتحركاً منها إلى اليسار.

وهذه الخاصية مفيدة في التقارير المالية وبعض أنواع التطبيقات العملية الأخرى.

توجد اختلافات عديدة في جملة PRINT USING حيث أن لها تفاصيل كثيرة جداً عرضها في هذا الكتاب غير عملي. وننصح القارئ أن يرجع في ذلك إلى الكتيب الخاص ببرمجة الحاسب الذي يفتنيه أو يستخدمه.

LPRINT and LPRINT USING

تتضمن بعض نسخ بيسك الحاسبات الدقيقة على جملة LPRINT وهي تستخدم في طباعة بيانات الإخراج على آلة الطباعة أو على نهاية طرفية طابعة (بدلاً من مراقبة ذلك على شاشة العرض TV). والجملة مطابقة تماماً لجملة PRINT ماعداً أننا نستخدم الكلمة الدالة LPRINT بدلاً من PRINT.

وهذه النسخ من بيسك للحاسبات الدقيقة والتي تستخدم جملة PRINT USING يمكن أيضاً أن تتضمن جملة LPRINT USING المماثلة.

مثال ٩ — ٢٦

يحتوى برنامج بيسك على جملي الطباعة التاليين :

```
200 PRINT A,B,C
210 LPRINT A,B,C
```

وسوف تسبب الجملة الأولى عرض قيم المتغيرات A و B و C على شاشة العرض TV ، بينما الجملة الثانية تسبب طباعة نفس القيم على نهاية طرفية طابعة .

General Comments تعليقات عامة

نذكر القارئ أن الجمل التي تم عرضها فيما سبق تهدف إلى أن تكون ممثلة لنسخ بيسك الحاسبات الدقيقة المتاحة عموماً . وقد تكون متاحة أو غير متاحة في نسخة معينة من اللغة ، وإذا كانت متاحة فقد تكون التفاصيل مختلفة نوعاً ما ، ومعظم نسخ بيسك الحاسبات الدقيقة تتضمن أيضاً جملاً إضافية لم تتم مناقشتها . ويحتوى ملحق (د) على ملخص أكثر شمولاً للجمل المتاحة عموماً في معظم نسخ ميكروسوفت بيسك . ويجب أن يرجع القارئ مرة أخرى إلى المرجع الخاص بالآلة التي يستخدمها لتفاصيل ومعلومات أكثر عن اللغة المستخدمة .

مثال ٩ — ٢٧ البحث عن نهاية عظمى على الحاسب الدقيق Search for a Maximum on a Microcomputer

نفرض أننا رجعنا إلى مثال البرمجة الذي اعتبرناه في مثال ٦—٦، والآن سنستخدم عدداً من الخصائص الجديدة المتاحة على ميكروسوفت بيسك كما هي منفذة على حاسب IBM الشخصي . ويمثل شكل ٩—٣ نسخة من برنامج الحاسب الدقيق . والمنطق المستخدم هو أساساً نفس العروض في مثال ٦—٦ رغم أن أسماء المتغيرات قد تم تمثيلها وتم استخدامها بعض البرامج المختلفة التي تمت كتابتها سابقاً ، وبالتحديد فإن المتغيرات التي سبق تسميتها X1,X2,X3,X4 أصبحت تسمى left,YL,YR,Right على الترتيب ، وأيضاً فإن Y2 و Y3 أصبحت تسمى YL و YR . و D أصبحت تسمى DISTANCE ، و I أصبحت تسمى COUNT% . وأسماء المتغيرات هذه تم تغييرها حتى تكون أكثر وصفاً لحقيقة المسألة . والبرنامج هو أكثر عمومية نوعاً ما عن ذلك الموضح في مثال ٦ — ٦ من حيث أن أكبر عدد من العمليات المكررة هي الآن متغير مدخل (MAXCOUNT%) بدلاً من مقدار ثابت موجود في البرنامج .

هذا بالإضافة إلى هذا البرنامج يحتوى على عدد من الجمل الجديدة التي سبق وصفها حالاً . وعلى الأخص فإننا نخص DEFSNG (سطر 30) IF—THEN الممتد (سطر 160) IF—THEN—ELSE المتداخلة (سطر 230) حلقه WHILE—WEND (سطر 150 إلى 240) ، امتداد استخدام جملة INPUT (سطر 70 إلى 100) وجملة PRINT USING (سطر 290 إلى 300) ، واستخدام هذه الجمل ينتج عنه برنامج أقصر وأكثر اختصاراً من البرنامج الموضح سابقاً في شكل (٦ — ٥) .

```

10 REM      SEARCH FOR A MAXIMUM OF THE FUNCTION  Y = X * COS(X)
20 '
30 DEFUNG A-Z
40 DEF FNY(X) = X*COS(X)
50 '
60 PRINT "Search for a Maximum of the Function  y = x * cos(x)" : PRINT
70 INPUT "Left boundary : ",LEFT
80 INPUT "Right boundary: ",RIGHT
90 INPUT "Minimum distance between interior points : ",DISTANCE
100 INPUT "Maximum number of iterations : ",MAXCOUNT% : PRINT
110 COUNT% = 0
120 '
130 REM      BEGIN LOOP
140 '
150 WHILE (RIGHT - LEFT) > 3 * DISTANCE
160     IF COUNT% = MAXCOUNT% THEN PRINT "Too many iterations" : END
170     COUNT% = COUNT% + 1
180     XL = LEFT + .5 * (RIGHT - LEFT - DISTANCE)
190     XR = XL + DISTANCE
200     YL = FNY(XL)
210     YR = FNY(XR)
220     GOSUB 360
230     IF YL = YR THEN 280 ELSE IF YL < YR THEN LEFT = XL ELSE RIGHT = XR
240 WEND
250 '
260 REM      COMPUTE AND PRINT FINAL SOLUTION
270 '
280 X = .5 * (XL + XR)
290 PRINT : PRINT "Xmax = " ; : PRINT USING "##.##### " ; X ;
300 PRINT SPC(12) ; "Ymax = " ; : PRINT USING "##.##### " ; FNY(X)
310 PRINT : PRINT "Number of iterations = " ; COUNT%
320 END
330 '
340 REM      SUBROUTINE TO PRINT THE RESULTS OF EACH ITERATION
350 '
360 PRINT : PRINT ,YL,YR
370 PRINT LEFT,XL,XR,RIGHT
380 RETURN
390 END

```

شكل ٩ - ٣

وهناك بعض الخصائص الأخرى لهذا البرنامج والتي لابد من الإشارة إليها . لاحظ أن العديد من السطور خالٍ إلا من الفاصلة العليا . وهذه الفاصلة العليا هي طريقة أخرى لكتابة ملاحظة في ميكروسوفت بيسك (كل ما يتبع الفاصلة العليا يعتبر ملاحظة) وفي المثال الحالي تكون هذه الملاحظات هي طريقة فقط لادخال سطور خالية حتى تفصل الأقسام المختلفة للبرنامج (لن يقبل المفسر سطوراً مرقمة خالية بالكامل) .

بالإضافة إلى ذلك يجب ملاحظة أن جملة END تظهر في ثلاثة أماكن مختلفة داخل البرنامج (سطر 390,320,160) وفي السطرين الأولين تستخدم جملة END بدلاً من جملة STOP حتى توقف عبارة غير مرغوب فيها تولدها STOP . وفي السطر الأخير تستخدم END بالطريقة التقليدية المعتادة ، ومن الغريب أنه لا توجد حاجة إلى END في نهاية برنامج ميكروسوفت بيسك . وعلى ذلك فإن ظهورها في نهاية البرنامج غير ضروري .

شكل ٩-٤ يبين المخرجات المولدة عند تنفيذ البرنامج باستخدام معالم الإدخال والتي هي أساساً نفسها المستخدمة في مثال ٦-٦ (رغم السماح لعدد أقل من العمليات المكررة هذه المرة ، حيث أننا نعرف عددها الذي نحتاج إليه) . وفي العادة فإن هذه المخرجات ستظهر على شاشة TV ولاحظ أن البيانات المدخلة تحتها خط .

ومن المفيد مقارنة هذه المخرجات بتلك الموضحة بشكل ٦-٦ .

Search for a Maximum of the Function $y = x * \cos(x)$

Left boundary : 0

Right boundary: 3.14159

Minimum distance between interior points : 0.0001

Maximum number of iterations : 20

	8.063497E-05		-7.647047E-05
0	1.570745	1.570845	3.14159
	.5553565	.5553716	
0	.7853725	.7854725	1.570845
	.4508655	.4507949	
.7853725	1.178059	1.178159	1.570845
	.5454382	.5454121	
.7853725	.9817156	.9818156	1.178159
	.5605342	.5605293	
.7853725	.8835441	.8836441	.9818156
	.5604048	.5604101	
.7853725	.8344583	.8345583	.8836441
	.5610945	.5610948	
.8344583	.8590012	.8591012	.8836441
	.5609718	.5609695	
.8590012	.8712727	.8713727	.8836441
	.5610725	.5610713	
.8590012	.865137	.865237	.8713727
	.5610933	.5610929	
.8590012	.8620691	.8621691	.865237
	.5610963	.5610963	
.8590012	.8605352	.8606352	.8621691
	.561096	.5610961	
.8590012	.8597682	.8598682	.8606352
	.5610963	.5610964	
.8597682	.8601517	.8602518	.8606352
	.5610963	.5610963	
.8601517	.8603434	.8604435	.8606352

Xmax = 0.860393

Ymax = 0.561096

Number of iterations = 14

شكل ٩ - ٤

٣-٩ دوال مكتبة إضافية ADDITIONAL LIBRARY FUNCTIONS

تتضمن معظم نسخ بيسك الحاسبات الدقيقة عدداً من الدوال المكتبة الخاصة ، بالإضافة إلى الدوال الموجودة أصلاً في النسخ التقليدية من اللغة . وقد سبق فعلاً أن ناقشنا حالتين جديدتين منها INPUT%-INKEY% في القسم ٢-٩ . والعديد من هذه الدوال الأكثر استخداماً في الحاسبات الدقيقة ملخصة فيما يلي.. لاحظ أن معظم هذه الدوال تمثداً بإمكانيات عديدة متعلقة بسلاسل الحروف .

مثال	الغرض	الدالة
$Y\# = CDBL(3*X1-2*Y1)$	تحول قيمة التعبير العددي e إلى مزدوج الدقة	$CDBL(e)$
$Y\% = CINT(3*X1-2*Y1)$	تحول (بالتقريب) قيمة التعبير العددي e إلى عدد صحيح	$CINT(e)$
$Y1 = CSNG(2*A\#/B\#)$	تحول قيمة التعبير العددي e إلى عدد مفرد الدقة .	$CSNG(e)$
$PRINT FRE(0)$	تعيد عدد البايت الغير مستخدمة في الذاكرة (n هو متغير وهمي)	$FRE(n)$
$Y\$ = INKEY\$$	تعيد حرف من لوحة المفاتيح (الحرف لن يتم عرضه) .	$INKEY\$$
$Y\$ = INPUT\(3) (تعيد سلسلة حروف من ٣ حروف) .	تعيد متغير سلسلة حروف من n حرف من لوحة المفاتيح (سلسلة الحروف لن يتم عرضها) .	$INPUT\$(n)$
$Y = INSTR(A\$,B\$)$ (إذا كان "BASIC" $A\$ = "BASIC"$) ($B\$ = "AS"$, $Y = 2$)	تعيد الموضع حيث توجد سلسلة حروف (s2) داخل سلسلة حروف أخرى (s1)	$INSTR(s1,s2)$
$Y\$ = LEFT\$(A\$,4)$ (إذا كان "COMPUTER" $A\$ = "COMPUTER"$) (فإن $Y\$ = "COMP"$)	تعيد n حرف في أقصى الشمال من سلسلة الحروف S	$LEFT\$(s,n)$
$Y\$ = MID\$(A\$,3,5)$ (إذا كان "COMPUTER" $A\$ = "COMPUTER"$) (فإن $Y\$ = "MPUTE"$)	تعيد جزء من سلسلة الحروف S بطول n من الحروف ابتداءً من الحرف الذي رقمه M	$MID\$(s,m,n)$
$Y = LEN(A\$)$ (إذا كان "COMPUTER" $A\$ = "COMPUTER"$) (فإن $Y = 8$)	تعيد عدد الحروف في سلسلة الحروف S	$LEN(s)$
$Y\$ = RIGHT\$(A\$,4)$ (إذا كان "COMPUTER" $A\$ = "COMPUTER"$) (فإن $Y\$ = "UTER"$)	تعيد n من الحروف في أقصى اليمين من سلسلة الحروف S	$RIGHT\$(s,n)$
$PRINT X;SPACES\$(5);Y$ (قيمتي X and Y سوف تفصل بخمسة مسافات فارغة)	تعيد سلسلة n من المسافات الفارغة	$SPACES\$(n)$
$Y\$ = STR\$(K+1)$ (إذا كان $K=5$ فإن $Y\$ = "6"$)	تعيد تمثيل سلسلة حروف من التعبير العددي e	$STR\$(e)$
$Y\$ = STRING\$(8,42)$ لأن $Y\$ = "*****"$ ($CHR\$(42) = "*"$)	تعيد متغير حرفي m حرف من سلسلة حروف لها كود ASCII هو n (n لا يمكن أن تزيد عن 255)	$STRING\$(m,n)$
$Y = VAL(N\$)$ (إذا كان "900" $N\$ = "900"$) (فإن $Y = 900$)	تعيد التمثيل العددي من سلسلة الحروف S بفرض أن S تتكون من اعداد مسبقة بإشارة إختيارية) .	VAL

وسوف تناقش بعض الدوال المكتبة الاضافية المستخدمة مع الحاسبات الدقيقة عند عرض موضوعات أخرى ، ويحتوى الملحق (د) على ملخص لكل الدوال المكتبة في ميكروسوفت بيسك المستخدمة عموماً .

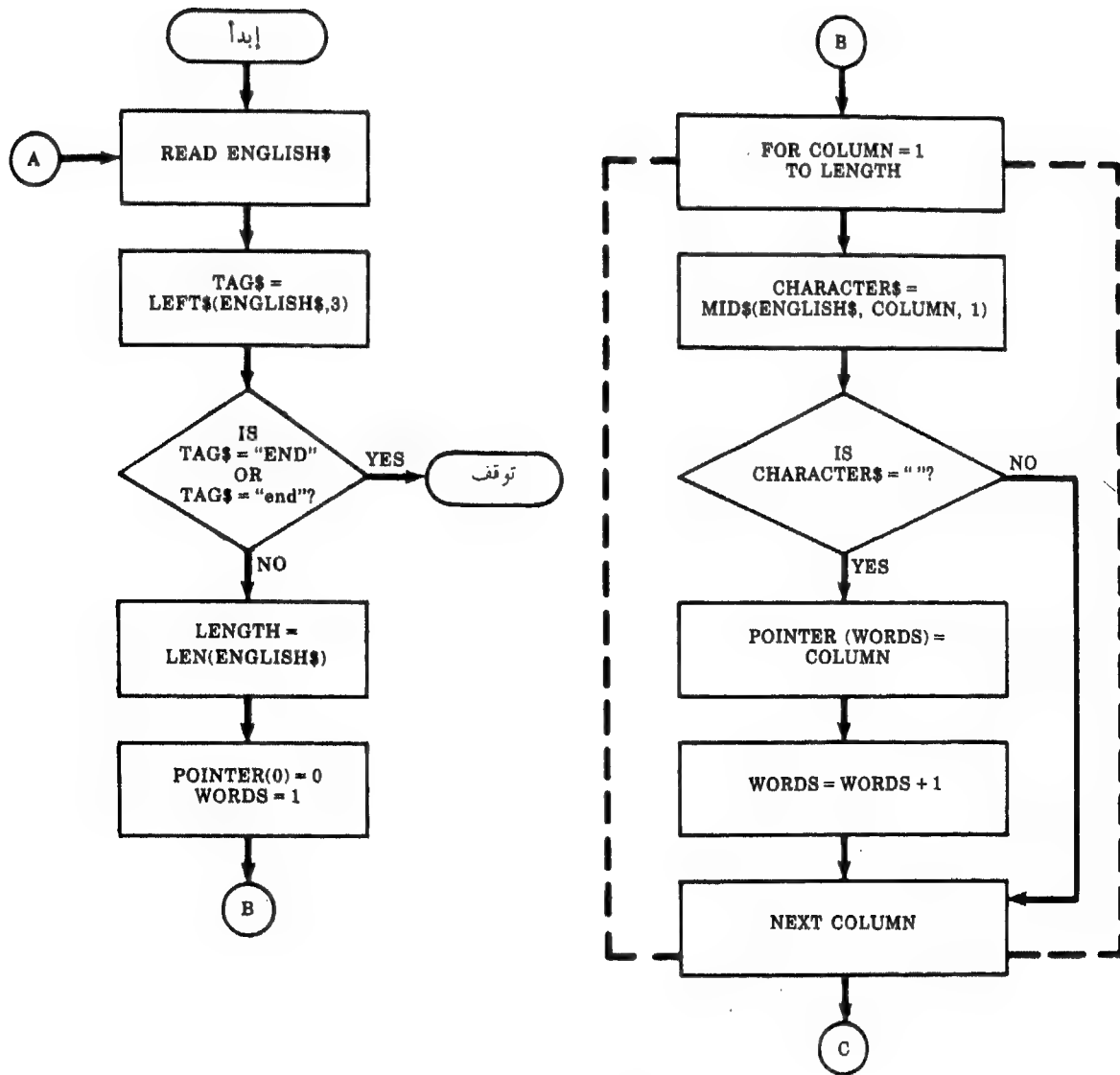
مثال ٩ - ٢٨ توليد بيجلاتين على حاسب دقيق Generating Piglatin on a Microcomputer

في مثال ٦-١٥ كتبنا برنامج بيسك لتوليد بيجلاتين من سطر في نص باللغة الإنجليزية . دعنا الآن نعيد هذا المثال مستخدمين العديد من الدوال الفريدة في ميكروسوفت بيسك .

المخطط التمهيدى للبرنامج . The Program Outline

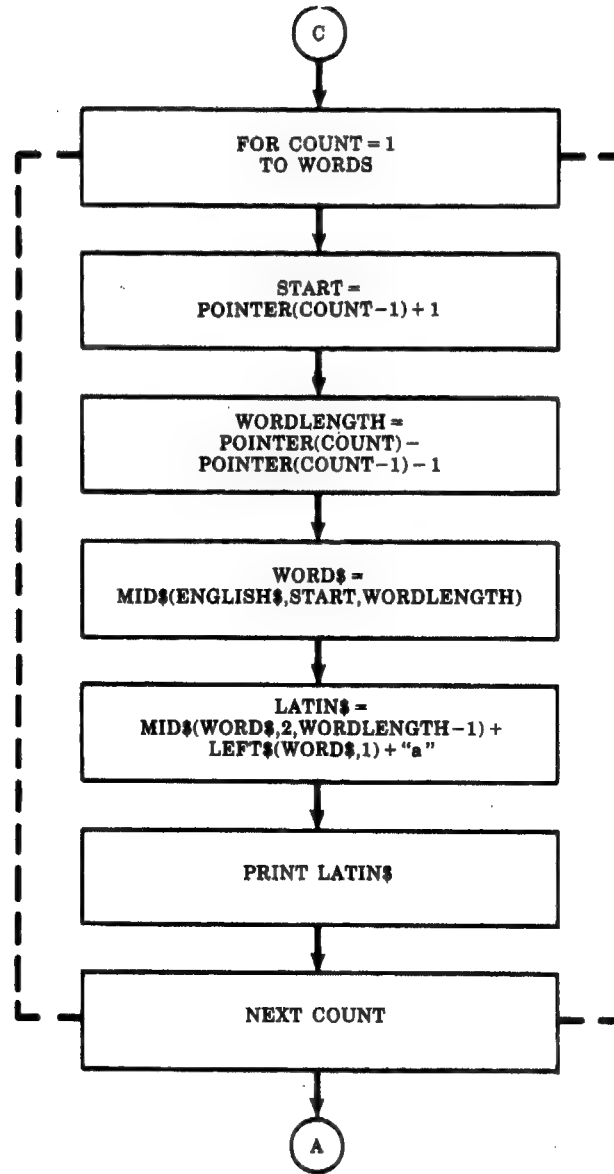
حتى يمكننا القيام بأحسن استخدام لهذه الدوال الجديدة سنستخدم استراتيجية مختلفة عن السابق استخدامها في مثال ٦-١٥ . وقبل وصف الطريقة الحسابية ، ستقوم بإدخال المتغيرات :

ENGLISH\$	=	سلسلة حروف للسطر المعطى بالنص باللغة الإنجليزية .
TAG\$	=	الحروف الثلاثة الأولى في السطر الأصلي من النص باللغة الإنجليزية .
POINTER(I)	=	صف من العناصر التى تمثل موقعاً في الفراغ بعد الكلمة التى ترتيبها I فى السطر المعطى من النص باللغة الإنجليزية .
LENGTH	=	عدد الحروف فى السطر المعطى من النص باللغة الإنجليزية .
WORDS	=	عدد الكلمات فى السطر المعطى من النص باللغة الإنجليزية .
COLUMN	=	عداد يبين موقع العمود فى السطر المعطى من النص باللغة الإنجليزية .
CHARACTERS\$	=	الحرف الموجود فى عمود معين فى السطر المعطى من النص باللغة الإنجليزية .
COUNT	=	عداد الكلمة للسطر من النص باللغة الإنجليزية .
WORDS\$	=	سلسلة حروف تمثل كلمة فى السطر المعطى من النص باللغة الإنجليزية .
START	=	رقم العمود الذى يبين البداية (الحرف الأول) للكلمة فى السطر المعطى من النص باللغة الإنجليزية .
WORDLENGTH	=	الطول (عدد الحروف) لكلمة فى السطر المعطى من النص باللغة الإنجليزية .
LATIN\$	=	سلسلة حروف تمثل بيجلاتين المكافئ لكلمة فى السطر المعطى من النص باللغة الإنجليزية .



شكل ٩ - ٥

تكملة في الصفحة التالية



شكل ٩ - ٥ (تكملة)

ويمكن وصف الحسابات الآن كما يلي :-

- ١ - اقرأ سطرًا من النص باللغة الانجليزية ، وخصص له ENGLISH \$
- ٢ - اختبر شروط الوقوف لتحديد ما إذا كانت الحروف الثلاثة الأولى من النص باللغة الإنجليزية مثل :

TAG\$ = LEFT\$(ENGLISH\$,3)

هي «END» أو «end» . وإذا كان كذلك ، فقم بإنهاء الحسابات ، وإلا فأكمل كالاتي

- ٣ - أوجد نهاية كل كلمة ، وقيم بعدد الكلمات باستخدام الطريقة التالية :

(أ) حدد موضع الفراغ التالي بعد كل كلمة مفردة على سطر النص باللغة الإنجليزية ، ثم خصص قيمته لـ POINTER، وعلى ذلك

فإن كل قيمة لـ POINTER ستحدد نهاية الكلمة المناظرة . فمثلاً (1) POINTER ستحدد نهاية الكلمة الأولى (بداية الكلمة الثانية) ، (2) POINTER ستحدد نهاية الكلمة الثانية (بداية الكلمة الثالثة) الخ ... لاحظ أن (0) POINTER ستخصص لها قيمة 0 مشيرة إلى بداية الكلمة الأولى وذلك مباشرة بعد العمود صفر .

(ب) كل مرة تجد فيها فراغاً ، زد عداد الكلمة (1) WORDS=WORDS+1 حيث WORDS خصص لها أساساً قيمة 1 .

٤ — استخراج كل كلمة من السطر الأصلي من النص وحولها إلى بيجلاتين ويمكن أن يتم ذلك كما يلي :-
(أ) ابدأ بموضع واحد بعد القيمة الأخيرة من POINTER إلى

$$START = POINTER(COUNT-1) + 1$$

(ب) حدد طول الكلمة كالآتي :-

$$WORDLENGTH = POINTER(COUNT) - POINTER(COUNT - 1) - 1$$

(ج) استخراج الكلمة الإنجليزية :

$$WORD\$ = MID$(ENGLISH$, START, WORDLENGTH)$$

(د) كون كلمة بيجلاتين المكافئة كالآتي :-

$$LATIN\$ = MID$(WORD$, 2, WORDLENGTH-1) + LEFT$(WORD$, 1) + "a "$$

٥ — اطبع كلمة البيجلاتين .

٦ — استمر حتى يتم تحويل كل الكلمات في السطر المعطى بالنص ويتم طبعها ، ثم ارجع ثانياً للخطوة ١ وابدأ من جديد . وبين شكل ٩-٥ خريطة سير عمليات هذه الطريقة .

برنامج اليبسك The BASIC Program

يحوي الشكل ٩-٦ برنامج اليبسك الحقيقي . لاحظ استخدام العديد من الدوال المكتوبة الجديدة وهي LEFT\$ (السطور 100 إلى 290) ، LEN (السطر 120) ، MID\$ (السطور 180, 280, 290) . لاحظ أيضاً استخدام عمليات التسلسل في السطر 290 . وفي النهاية لاحظ أن ملاحظات البرنامج تستخدم كلها الفاصلة العليا بدلاً من جملة REM (السطور 10, 60, 140, 230) .

وطول هذا البرنامج تقريباً هو نفس طول مولد بيجلاتين المكتوب بلغة اليبسك التقليدية ، كما هو موضح في شكل ٦-٩ (كجزء من مثال ٦-١٥ . وعلى ذلك فإن استخدام الخصائص الجديدة للحسابات الدقيقة لم ينقص من طول البرنامج بشكل مغنوى . والمنطق خلف البرنامج الخالى هو أنه أكثر مباشرة للمثال . ولا بد للقارئ أن يقوم — بحرص — بمقارنة البرنامجين على هذا الأساس .

والشكل ٩-٧ يبين المخرج الذى يتولد عند تنفيذ البرنامج . لاحظ أن النص — باللغة الإنجليزية — الذى تم ادخاله كمدخل هو نفسه كما في مثال ٦-١٥ . باستثناء استخدام كل الحروف العليا والمنخفضة (البيانات المدخلة مرة أخرى تحتها خط) ولا بد للقارئ أن يقارن شكل ٩-٧ و ٦-١٠ ليرى تناظرهما .


```

10 '*** MICROCOMPUTER PIGLATIN GENERATOR ***
20 '
30 DIM POINTER(80)
40 PRINT "Welcome to Microcomputer Piglatin" : PRINT
50 '
60 '*** READ A LINE OF ENGLISH TEXT
70 '
80 PRINT "Enter a line of text below: "
90 INPUT "",ENGLISH$
100 TAG$ = LEFT$(ENGLISH$,3)
110 IF TAG$ = "END" OR TAG$ = "end" THEN END
120 LENGTH = LEN(ENGLISH$)
130 '
140 '*** FIND THE END OF EACH WORD AND COUNT THE NUMBER OF WORDS
150 '
160 POINTER(0) = 0 : WORDS = 1
170 FOR COLUMN = 1 TO LENGTH
180   CHARACTER$ = MID$(ENGLISH$,COLUMN,1)
190   IF CHARACTER$ = " " THEN POINTER(WORDS) = COLUMN : WORDS = WORDS + 1
200 NEXT COLUMN
210 POINTER(WORDS) = LENGTH + 1
220 '
230 '*** CONVERT EACH WORD TO PIGLATIN AND PRINT
240 '
250 FOR COUNT = 1 TO WORDS
260   START = POINTER(COUNT-1) + 1
270   WORDLENGTH = POINTER(COUNT) - POINTER(COUNT-1) - 1
280   WORD$ = MID$(ENGLISH$,START,WORDLENGTH)
290   LATIN$ = MID$(WORD$,2,WORDLENGTH-1) + LEFT$(WORD$,1) + "a "
300   PRINT LATIN$;
310 NEXT COUNT
320 PRINT : PRINT
330 GOTO 80
340 END

```

شکل ۱ - ۶

Welcome to Microcomputer Piglatin

Enter a line of text below:
this is a piglatin generator
 hista sia aa iglatinpa eneratorga

Enter a line of text below:
WHAT SORT OF GARBLED MESSAGE IS THIS ANYHOW
 HATW_a ORTS_a FO_a ARBLEDB_a ESSAGEM_a SI_a HIST_a NYHOWA_a

Enter a line of text below:
Now is the time for all good men to come to the aid of their country
 owNa sia heta imeta orfa llaa oodga enma ota omeca ota heta idaa foa heirta

ountryca

Enter a line of text below:
end

شکل ۱ - ۷

٩ - ٤ ملفات البيانات للحاسبات الدقيقة MICROCOMPUTER DATA FILES

تعرض أهم التطبيقات على الحاسب الدقيق لاستخدام ملفات بيانات تحت سيطرة برنامج بيسك . وتشجع معظم نسخ بيسك الحاسبات الدقيقة كلا من ملفات البيانات المتسلسلة والمشوائية ، بالرغم من وجود اختلافات جديرة بالاهتمام في طريقة التوصل إلى هذه الملفات من نسخة بيسك إلى أخرى . وعلاوة على ذلك ، إجراءات إدارة البيانات المستخدمة للحاسبات الدقيقة عموماً ليست هي نفسها كإجراءات إدارة البيانات المستخدمة على الحاسبات الكبيرة التي قدمت في الفصل الثامن . (لاحظ أن استخدام ملفات بيانات الحاسبات الدقيقة يتطلب نوعاً من أجهزة التخزين المساعدة ، مثل شريط كاسيت مغناطيسي ، والقرص المرن والقرص الصلب ، أو جزء الذاكرة من أشباه الموصلات) .

وبعد في هذا القسم عدة أمثلة بسيطة توضح استخدام ملفات البيانات باستخدام نسخة ميكروسوفت بيسك الموجودة على حاسب IBM الشخصي ، ولكن يجب أن يفهم القارئ أن هذه الأمثلة هي فقط لتطبيقات ملف بيانات الحاسب الدقيق . ويجب أن يستشير مخطط البرامج الكتيب الخاص بحاسبه الدقيق ليقرر إجراءات إدارة الملف المناسبة .

ملفات البيانات المتسلسلة Sequential Data Files

مع معظم الحاسبات الدقيقة ، يتكون ملف البيانات المتسلسلة من مجموعة مضاعفة من بنود البيانات ، ومرتبطة بالتسلسل ، بدون اعتبار لأرقام السطور (وهذا يختلف عن ملفات البيانات المتسلسلة الموصوفة في الفصل الثامن حيث أرقام السطور جزء مكرر للبيانات) . تنشأ مثل هذه الملفات ثم تقرأ وتمثل تحت تحكم البرنامج مباشرة . والطريقة التي تنجز بها موضحة في المثال التالي :

مثال ٩ - ٢٩ إنشاء ملف بيانات متسلسل Creating a Sequential Data File

يبين شكل ٩ - ٨ برنامج بيسك للحاسب IBM الشخصي لإنشاء ملف بيانات متسلسل يحتوي على أسماء الطلبة ودرجات اختباراتهم المبينة في شكل ٨ - ١ (لاحظ الآن أن ملف البيانات المتسلسل يجب أن تخلق بواسطة برنامج بيسك بدلاً من طباعتها مباشرة إلى الحاسب كما هو موصوف في الفصل الثامن) .

السطر 30 في هذا البرنامج يتسبب في فتح ملف البيانات المتسلسل المسمى SCORES في حالة الخرج ("O") وتحدد له قناة البيانات رقم 1 . والسطران 40 و50 ينمجان لسلسلي الحروف TITLES و TERMS بالدخول من لوحة المفاتيح : ثم بعد ذلك تكتب حروف هاتان السلسلتان الحرفيتان على ملف البيانات في السطرين 60 و70 . وبالمثل.. فنقل البيانات يحدث في السطور 80 و110 و120 ويستمر تكرار هذا الإجراء حتى تدخل كلمة END بدلاً من اسم الطالب . وبعد ذلك يقفل ملف البيانات وتنتهي الحسابات .

وعند تشغيل البرنامج فإن عنوان السنة الدراسية والفصل الدراسي (مثل 100 COMPUTER SCIENCE و FALL 1982) يتم إدخالهما في سطور منفصلة ويتبعهما رجوع عربة الآلة . وبالمثل بالنسبة لأسماء الطلبة .. كل اسم طالب يتبعه خمس درجات للاختبارات على سطر منفصل . والدرجات الخمس كلها تدخل على نفس السطر تفصل بين كل درجة أخرى فصلة (,) وأخير درجة يتبعها رجوع عربة الآلة .

يبين شكل ٩ - ٩ محتويات ملف البيانات ينتج عند تشغيل هذا البرنامج باستخدام أسماء الطلاب ودرجات الاختبارات المعطاه في شكل ٨ - ١ . ملف البيانات هذا سوف يسمى SCORES . قارن شكل ٩ - ٩ على ملف البيانات المناظر والمبين في أسفل شكل (٨ - ١) . يجب أن يفهم أن هذه المعلومات سوف تكتب على وسيلة تخزين مساعدة (مثل القرص المرن) بدلاً من آلة الطباعة . وبمجرد أن يتم إنشاء ملف البيانات المتسلسل فإنه يمكن وضع محتوياته كقائمة على آلة الطباعة ، وهذا يبين كيف تم توليد شكل ٩ - ٩ .

والإجراء العام لتعديل ملف البيانات هو نسخ محتويات ملف البيانات القديم على ملف البيانات الجديد مع دمج أى إضافات أو تغييرات فى البيانات أثناء إجراء النسخ . وبعد الانتهاء من عملية التعديل يلغى الملف القديم "KILLED" ويمطى ملف البيانات الجديد (المعدل) اسم الملف القديم .

```

10 REM ***** CREATE A SEQUENTIAL DATA FILE *****
20 REM      (STUDENT EXAMINATION SCORES)
30 OPEN "0",1,"SCORES"
40 INPUT "Course title";TITLE$
50 INPUT "Term";TERM$
60 PRINT #1,TITLE$
70 PRINT #1,TERM$
80 PRINT: INPUT "Name";N$
90 IF N$="END" OR N$="end" THEN 140
100 INPUT "Exam scores";C1,C2,C3,C4,C5
110 PRINT #1,N$
120 PRINT #1,C1,C2,C3,C4,C5
130 GOTO 80
140 CLOSE
150 END

```

شكل ٩ - ٨

```

Comp Sci 141
Fall 1985
Adams B F
  45 80 80 95 55
Brown P
  60 50 70 75 55
Davis R A
  40 30 10 45 60
Fisher E K
  0 5 5 0 10
Hamilton S P
  90 85 100 95 90
Jones J J
  95 90 80 95 85
Ludwig C W
  35 50 55 65 45
Osborne T
  75 60 75 60 70
Prince W F.
  85 75 60 85 90
Richards E N
  50 60 50 35 65
Smith M C
  70 60 75 70 55
Thomas B A
  10 25 35 20 30
Wolfe H
  25 40 65 75 85
Zorba D R
  65 80 70 100 60

```

شكل ٩ - ٩

```

10 REM ***** PROGRAM TO PROCESS STUDENT EXAMINATION SCORES *****
20 REM *****          USING SEQUENTIAL DATA FILES          *****
30 DIM C(15)
40 OPEN "I",1,"SCORES"
50 OPEN "O",2,"UPDATE"
60 INPUT #1,TITLE$
70 INPUT #1,TERM$
80 PRINT "Course title: ";TITLE$,"Term: ";TERM$
90 PRINT #2,TITLE$: PRINT #2,TERM$
100 PRINT: INPUT "Exam number";K
110 PRINT: INPUT "Calculate averages (Y/N) ";ANS$
120 PRINT: INPUT #1,N$
130 FOR I=1 TO K-1: INPUT #1,C(I): NEXT I
140 PRINT N$,
150 INPUT "New score";C(K)
160 PRINT #2,N$
170 SUM=0
180 FOR I=1 TO K: PRINT #2,C(I);: SUM=SUM+C(I): NEXT I
190 IF ANS$="N" OR ANS$="n" THEN 230
200 AVG=SUM/K
210 PRINT "Average=";AVG
220 PRINT #2,AVG
230 IF NOT EOF(1) THEN 120
240 CLOSE
250 KILL "SCORES"
260 NAME "UPDATE" AS "SCORES"
270 END

```

شکل ٩ - ١٠

```

Course title: Comp Sci 141 Term: Fall 1985
Exam number? 6
Calculate averages (Y/N) ? y
Adams B F      New score? 75
Average= 71.66666
Brown P       New score? 80
Average= 65
Davis R A     New score? 55
Average= 40
Fisher E K    New score? 5
Average= 4.166667
Hamilton B P  New score? 90
Average= 91.66666
Jones J J     New score? 80
Average= 87.5
Ludwig C W   New score? 70
Average= 53.33333
Osborne T    New score? 80
Average= 70
Prince W F   New score? 100
Average= 82.5
Richards E N New score? 70
Average= 55
Smith M C    New score? 75
Average= 67.5
Thomas B A   New score? 10
Average= 21.66667
Wolfe H      New score? 95
Average= 64.16666
Zorba D R    New score? 95
Average= 78.33334

```

شکل ٩ - ١١

مثال ٩ - ٣٠ تشغيل درجات اختبارات طالب على حاسب دقيق

Processing Student Examination Scores on a Microcomputer

نرى في شكل ٩ - ١ برنامج بيسك خاص بالحاسب الدقيق راديو شاك في . آر . إس 80 يسمح لنا بتعديل ملف البيانات المتسلسل الذي تم إنشاؤه في المثال ٩ - ٢٩ ، (وهذا البرنامج يوازي برنامج بيسك للحاسب الكبير المين في شكل ٨ - ٥ ، والذي يحقق نفس الغرض) .

وفي هذا البرنامج يتسبب السطر 40 في فتح ملف البيانات القديم المسمى SCORES في حالة إدخال الخط ("I") وإعطائه قناة البيانات (الخزن الوسيط) رقم 1 . وبالمثل، فالسطر 50 يتسبب في فتح الملف الجديد المسمى UPDATE في حالة إخراج الخط ("O") وإعطائه قناة البيانات رقم 2 . ثم بعد ذلك يقرأ عنوان المادة الدراسية (TITLE\$) والفصل الدراسي (TERM\$) من الملف SCORES (السطران 60 و 70) ثم تعرض على شاشة مراقب TV (السطر 80) وتكتب أيضاً إلى الملف UPDATE (السطر 90) يسمح السطران 100 و 110 للمستفيد أن يصف رقم الاختبار المراد حساب متوسط الدرجات لكل طالب أم لا . ثم بعد ذلك نقابل حلقة تكرارية تتراوح ما بين السطور 120 إلى 240 حيث يتم إجراء الخطوات التالية لكل طالب :

- ١ - تقرأ اسم الطالب ودرجات الاختبار السابق SCORES (السطران 120 و 130) .
- ٢ - يعرض اسم الطالب على شاشة المراقبة TV (السطر 140) ثم يتم إدخال درجة جديدة من خلال لوحة المفاتيح (السطر 150)
- ٣ - يكتب اسم الطالب ودرجات اختبار الطالب على UPDATE ثم يحسب متوسط الدرجات (اختيارياً) (السطور 160 إلى 200)
- ٤ - يعرض المتوسط المحسوب على شاشة المراقبة TV (السطر 210) ثم يكتب إلى UPDATE (السطر 220) .
- ٥ - ويستمر في هذه العملية حتى تكتشف نهاية الملف SCORES (السطر 230) . وفي هذا الوقت نقفل الملفين (السطر 240) ، ويلغى الملف الأصلي (KILLED في السطر 250 . وسنسمى الملف الجديد SCORES (السطر 260) .

ملفات البيانات العشوائية Random Data Files

من وجهة نظر البرمجة .. فإن ملفات البيانات العشوائية أصعب في استخدامها شيئاً ما عن الملفات المتسلسلة ، رغم أنه يمكن التوصل إليها أسرع عند تنفيذ البرنامج. ومن ناحية أخرى فإن مركبات الملفات الفردية يمكن تداولها أسرع كثيراً في ملف التداول العشوائي كما هو موضح في الفصل الثامن .

وإجراءات إنشاء وقراءة وتعديل ملفات البيانات العشوائية تختلف بصورة عامة في الحاسب الدقيق عنها في الحاسب الكبير . فمثلاً على الحاسب IBM الشخصي يعرف ملف البيانات العشوائي كمجموعة من سجلات ذات طول ثابت . تخزن كل بنود البيانات الفردية في السجل كسلسلة حرفية ويجب تحويل بنود البيانات التي تمثل قيماً عديدة إلى أرقام حقيقية عند نقلها لذاكرة الحاسب . وبالمثل ، يجب تحويل هذه البنود مرة أخرى إلى سلاسل حرفية قبل نقلها إلى ملف البيانات . ينشأ مخزن وسيط يحتوي على سجل واحد في ذاكرة الحاسب . ويقسم المخزن الوسيط إلى حقول حيث يحوى كل حقل بنداً واحداً من بنود البيانات .

مثال ٩ - ٣١ التحكم في المخزون للحاسبات الدقيقة : Microcomputer Inventory Control

يحتوى شكل ٩ - ١٣ برنامجاً بسيطاً للتحكم في المخزون وهو مكتوب للحاسب IBM الشخصي، والطريقة العامة لهذه المسألة هي أساساً نفس الخطط التمهيدى للمثال ٨ - ١٠ ، يتشابه البرنامج لدرجة كبيرة مع النسخة المقدمة للحاسب الكبير في شكل ٨ - ١٢ ومع كل فقد تم إعادة تسمية المتغيرات لتكون أكثر وصفاً للمسألة الحقيقية وعلى الأخص رقم السجل ، يشار إليه الآن %RECNO وتسمى السلسلة الحرفية التي تمثل مستوى المخزون (أى عدد الوحدات) %UNITSS وعددها الصحيح المناظر يسمى %UNITS، ويسمى التغير في مستوى المخزون الآن %CHANGE.

Comp Sci 141
 Fall 1985
 Adams B F
 45 80 80 95 55 75 71.66666
 Brown P
 60 50 70 75 55 80 65
 Davis R A
 40 30 10 45 60 55 40
 Fisher E K
 0 5 5 0 10 5 4.166667
 Hamilton S P
 90 85 100 95 90 90 91.66666
 Jones J J
 95 90 80 95 85 80 87.5
 Ludwig C W
 35 50 55 65 45 70 53.33333
 Osborne T
 75 60 75 60 70 80 70
 Prince W F
 85 75 60 85 90 100 82.5
 Richards E N
 50 60 50 35 65 70 55
 Smith M C
 70 60 75 70 55 75 67.5
 Thomas B A
 10 25 35 20 30 10 21.66667
 Wolfe H
 25 40 65 75 85 95 64.16666
 Zorba D R
 65 80 70 100 60 95 78.33334

شکل ۹ - ۱۲

```

10 REM *** Microcomputer Inventory Control Program ***
20
30 OPEN "R",#1,"INVTRY",5
40 FIELD #1,5 AS UNITS%
50 PRINT "Stock numbers run from 1 to 2000"
60 PRINT: PRINT "To end session, enter a negative stock number"
70 PRINT: INPUT "Stock number";RECNO%
80 IF RECNO% < 1 THEN GOTO 260
90 IF RECNO% > 2000 THEN PRINT: GOTO 50
100
110 REM *** UPDATE A RECORD
120
130 GET 1,RECNO%
140 UNITS%=CVI(UNITS%)
150 PRINT "Original inventory=";UNITS%;" items"
160 INPUT "Change in inventory level";CHANGE%
170 UNITS%=UNITS%+CHANGE%
180 IF UNITS% < 0 THEN UNITS% = 0
190 PRINT "New inventory=";UNITS%;" items"
200 LSET UNITS%=MKI$(UNITS%)
210 PUT 1,RECNO%
220 GOTO 70
230
240 REM *** END RECORD UPDATE
250
260 CLOSE
270 END

```

شکل ۹ - ۱۲

Stock numbers run from 1 to 2000

To end session, enter a negative stock number

Stock number? 1186
Original inventory= 346 items
Change in inventory level? -45
New inventory= 301 items

Stock number? 708
Original inventory= 368 items
Change in inventory level? 200
New inventory= 568 items

Stock number? 84
Original inventory= 147 items
Change in inventory level? 16
New inventory= 163 items

Stock number? 1400
Original inventory= 78 items
Change in inventory level? -50
New inventory= 28 items

Stock number? -999

شكل ٩ - ١٤

٩ - ٥ إجراءات لغة الآلة في البيسك MACHINE-LANGUAGE PROCEDURES IN BASIC

تتضمن بعض نسخ بيسك الحاسبات الدقيقة إمكانيات للقيام بعمل بعض الإجراءات التي تلازم عادة لغة الآلة . وتتضمن هذه الإجراءات :

- ١ - اختيار جزء من ذاكرة الحاسب .
- ٢ - فحص محتويات ذاكرة الحاسب .
- ٣ - تغيير محتويات ذاكرة الحاسب .
- ٤ - التوصل إلى برامج فرعية بلغة الآلة .

المناقشة التفصيلية لاستخدام هذه الخصائص خارج نطاق وحدود هذا الكتاب ، إلا أن هذه الإجراءات تستخدم أساساً في تطبيقات مثل بيانات متقدمة ، وتوليد الصوت والموسيقى ، ومداولة الخطأ ، والتحكم في أجهزة الوحدات الخارجية .

والخواص الأكثر شيوعاً ملخصة فيما يلي .

الجملة أو الدالة	الفرع	مثال
DEF SEG=M	يختار جزء في الذاكرة ويبدأ في المكان M	10 DEF SEG=32768 (يختار جزء في الذاكرة ويبدأ في المكان 32768)
PEEK (M)	تعاود محتويات المكان M في الذاكرة (عشرى) .	10 LET Z=PEEK (768) (تحدد قيمة Z بمحتويات المكان رقم 768)
POKE M, X	توضع قيمة X في مكان الذاكرة M (عشرى)	20 POKE 200, 333 (توضع القيمة 333 في مكان الذاكرة رقم 200)
CALL V	يتوصل للبرنامج الفرعى بلغة الآلة والذي يبدأ في المكان رقم V من الذاكرة (عشرى)	20 START=1024 30 CALL START يتوصل للبرنامج الفرعى الذى يبدأ في المكان رقم 1024 من الذاكرة)
USR (X)	تُنقل قيمة X إلى البرنامج الفرعى بلغة الآلة .	40 PRINT USR (12) (تطبع قيمة مولدة بواسطة برنامج فرعى بلغة الآلة، الذى يقبل الرقم 12 كعامل إدخال) .

لاحظ : أن بعض نسخ البيسك تسمح بتوصيف بداية برنامج فرعى بلغة الآلة بواسطة جملة DEF ، فمثلاً 50 DEF USR = 1280 ، ونذكر القارئ أن CALL و DEF SEG, POKE و USR عبارة عن جمل بينما PEEK و USR عبارة عن دوال مكتنية .

مثال ٩ - ٣٢

ادرس جملة البيسك المبينة فيما يلي :

```
200 IF PEEK(768)> 127 THEN POKE 768,127
210 CALL START
```

تتسبب الجملة الأولى (السطر 200) في فحص محتويات المكان رقم 768 من الذاكرة (عشرى) واستبداله بالقيمة 127 إذا تعدت قيمته الأصلية الرقم 127 . وتتصل الجملة الثانية (السطر 210) بالبرنامج الفرعى بلغة الآلة الذى يبدأ في المكان START . (لاحظ أن التنفيذ الناجع لهذا البرنامج الفرعى يتطلب أن تكون القيمة المخزنة في المكان 768 لاتتعدى الرقم 127) .

٩ - ٦ خواص مستبعدة عن بييسك الحاسبات الدقيقة .

FEATURES EXCLUDED FROM MICROCOMPUTER BASIC

نذكر القارئ مرة أخرى بالتعبيرات في نسخ بييسك الحاسبات الدقيقة المختلفة المتوفرة حالياً . وبالرغم من أن معظم النسخ معقدة عمماً إلا أنه يوجد قليل من الخصائص التقليدية للبييسك التى لا توجد عادة في بييسك الحاسبات الدقيقة . وهذه تتضمن :

١ - جمل (المصفوفات) MAT (انظر الفصل السابع)

- ٢ - الدوال الجديدة السطور (انظر القسم ٦ - ٣) .
- ٣ - أوامر معينة للنظام (مثال OLD و REPLACE و SCRATCH) .
- ٤ - تفسين أسطر خالية (رغم أن بعض نسخ اللغة تسمح بسطر يحتوى على رقم الجملة متبوعة بواسطة فاصل ، مثال ، : 100) .
- ويجدر بنا أن نذكر القارئ أن البيسك ينمو ويتغير بسرعة وتزداد شعبيته كما أن الحاسبات الدقيقة تتزايد وتتكاثر بسرعة في المنازل ، المدارس والمكتبات في مجتمعاتنا . ولذا نتوقع خصائص جديدة متطورة في النسخ المستقبلية من اللغة .

اسئلة للمراجعة

Review Questions

- ١ - ٩ ماهي الميزة الموجودة من استخدام أسماء المتغيرات الطويلة ؟
- ٢ - ٩ هل يمكن استخدام كلمات ببسك الدالة (مثل PRINT و DATA و NEXT) كأسماء متغيرات ؟
- ٣ - ٩ اذكر أنواع البيانات المختلفة المتاحة في نسخ البيسك للحاسبات الدقيقة . كيف تميز بين أنواع المتغيرات المناظرة عن بعضها ؟
- ٤ - ٩ ماهو نوع النتيجة التي نحصل عليها من عملية رياضية تتضمن :
- (أ) بيانات صحيحة وحقيقية ؟
- (ب) بيانات صحيحة ومزدوجة الدقة ؟
- (ج) بيانات حقيقية ومزدوجة الدقة ؟
- ٥ - ٩ ماذا يحدث عندما تملأ بيانات رقمية من نوع ما إلى متغير رقمي من نوع مختلف ؟
- ٦ - ٩ ماذا يحدث إذا أعطينا كمية حقيقية أو مزدوجة الدقة إلى متغير صحيح ؟
- ٧ - ٩ كيف تختلف القسمة الصحيحة عن القسمة العادية ؟
- ٨ - ٩ ماهو الغرض من المعامل MOD ؟ ومع أي نوع من البيانات يجب أن يستخدم ؟
- ٩ - ٩ ماهو الغرض من المعاملات AND و OR و NOT ؟ وفي أي نوع من الجمل تستخدم ؟
- ١٠ - ٩ لخص التدرج الهرمي للمعاملات الحسابية والمنطقية والمتراصلة . هل كل نسخ البيسك للحاسبات الدقيقة تستخدم هذا التدرج الهرمي بالذات ؟
- ١١ - ٩ ماذا يقصد بالوصل ؟ وإلى أي نوع من البيانات نطبق الوصل ؟ وما هو المعامل الذي نستخدمه دائماً لنشير إلى الوصل ؟
- ١٢ - ٩ ماهي المزايا في كتابة جملتين أو أكثر على نفس السطر ؟ وكيف تميز هذه الجمل عن بعضها ؟
- ١٣ - ٩ ماهو الغرض من جمل DEFINT و DEFSNG و DEFDBL و DEFSTR ؟ وكيف تقارن استخدام هذه الجمل باستخدام الملاحق في أسماء المتغيرات
- ١٤ - ٩ صف الاستخدام الموسع لجملة IF-THEN في معظم نسخ البيسك المحسنة .
- ١٥ - ٩ صف جملة IF-THEN-ELSE وقارن استخدامها باستخدام جملة IF-THEN الموسعة . وما هو الشيء المنوي الذي أضافته عبارة ELSE ؟

- ١٦-٩ ماهو المقصود بالبرمجة الهيكلية ؟
- ١٧-٩ ماهو الغرض من جملة ON-GOSUB ؟ وكيف تختلف عن جملة ON-GO TO ؟
- ١٨-٩ ماهو الغرض من جملة ON ERROR GO TO ؟ وماهى الجملة الأخرى المستخدمة مقترنة بجملة ON ERROR GO TO ؟
- ١٩-٩ صف جملتي WHILE و WEND . وما هو النوع الجديد من البرنامج الذى يمكن أن يعرف بواسطة هاتين الجملتين ؟
- ٢٠-٩ اشرح الاستخدام لجملة INPUT فى معظم نسخ البيسك المحسنة .
- ٢١-٩ اشرح الاستخدام لكل من الدوال INPUT\$, INKEY\$ لخلق وقفات فى تنفيذ البرنامج . وهل كل نسخ البيسك الحاسبات الدقيقة تستخدم هاتين الدالتين ؟
- ٢٢-٩ تخضع أكثر الخصائص المشتركة الموجودة فى جملة PRINT USING . وكيف تنفذ هذه الخصائص ؟ وإلى أى حد يكون لهذه الخصائص معنى ؟
- ٢٣-٩ ماهو الغرض من جمل LPRINT و LPRINT USING ؟
- ٢٤-٩ صف الغرض من كل من الدوال المكتوبة التالية : CDBL, CINT, CSNG, FRE, INKEY\$, INPUT\$, INSTR, LEFT\$, LEN, MID\$, RIGHT\$, STR\$, STRING\$, VAL . كم عدد الخلاصات المطلوبة لكل دالة ؟ وما هو نوع الخلاصات المطلوبة ؟
- ٢٥-٩ ماهو الفرق بين ملفات البيانات عشوائية والمتسلسلة كما تنفذ على الحاسب الدقيق ؟ وماهى مزايا وعيوب كل منها ؟
- ٢٦-٩ كيف يختلف استخدام ملفات البيانات المتسلسلة على الحاسب الدقيق عن استخدام ملفات البيانات المتسلسلة على حاسب كبير (كما هو موصوف فى الفصل الثامن) ؟
- ٢٧-٩ كيف يختلف استخدام ملفات البيانات العشوائية على الحاسب الدقيق عن استخدام ملفات البيانات العشوائية على حاسب كبير (كما هو موصوف فى الفصل الثامن) ؟
- ٢٨-٩ صف الغرض لكل من الجمل أو الدوال التالية : DEF SEG, PEEK, POKE, CALL, USR . ماهى الجمل وماهى الدوال ؟ وفى أى نوع من التطبيقات تستخدم هذه الجمل والدوال .
- ٢٩-٩ اذكر أسماء خواص البيسك « القياسى » أو المعيارى، (المنفذ على الحاسبات الكبيرة) وغير المتاحة فى معظم نسخ بييسك الحاسبات الدقيقة .

مسائل تكميلية

Supplementary Problems

- « المسائل » التالية متعلقة بجميع المعلومات أكبر منها لحل مسائل حقيقة . أجب عن الأسئلة التى تخص نسخة بييسك الحاسبات الدقيقة التى تتعلق بالحاسب الخاص بك .
- ٣٠-٩ هل تقبل نسخة البيسك الخاصة بك استخدام أسماء المتغيرات الطويلة ؟ وإن كان ذلك صحيحاً ، ماهو أقصى طول مسموح به لكل من أسماء المتغيرات ؟
- ٣١-٩ ماهى أنواع البيانات الرقمية المسموح بها ؟ وماهو الاختلاف الموجود بين الأنواع المختلفة من المتغيرات الرقمية ؟ هل يمكن لكل المتغيرات المشتركة فى الحرف الأول أن تعرف وتكون من نوع معين من البيانات ؟

- ٣٢ - ٩ هل يمكن القيام بعمليات حسابية بين أنواع مختلفة من بيانات رقمية ؟ وما هو نوع النتيجة التي نحصل عليها مع كل تركيبه للبيانات ؟
- ٣٣ - ٩ هل يمكن إعطاء نوع بيانات معين لمتغير رقمي من نوع آخر ؟ وما هي نوع النتيجة التي نحصل عليها ؟
- ٣٤ - ٩ ما هي المعاملات الإضافية المتاحة ؟ وما هو الغرض من كل ؟
- ٣٥ - ٩ ما هو التدرج الهرمي الكامل للمعاملات الحسابية والمترابطة والمنطقية ؟
- ٣٦ - ٩ هل تقبل نسخة البيسك الخاصة بك الوصل ؟ وإذا كان ذلك صحيحاً ، ما هو المعامل المستخدم لهذا الغرض ؟
- ٣٧ - ٩ هل يمكن وضع عدة جمل على سطر واحد ؟ وكيف يمكن فصل هذه الجمل من بعضها ؟
- ٣٨ - ٩ كيف تنفذ جملة IF-THEN على نسخة البيسك الخاصة بك ؟ هل يمكن أن تتبع THEN عدة جمل ؟ وهل عبارة ELSE متاحة ؟
- ٣٩ - ٩ هل جملة ON-GOSUB متاحة على نسخة البيسك الخاصة بك ؟
- ٤٠ - ٩ هل جملة ONERROR GOTO متاحة على نسخة البيسك الخاصة بك ؟ وهل جملة RESUME متاحة ؟
- ٤١ - ٩ هل تسمح نسخة البيسك الخاصة بك خاصية التكرار المشروط ؟ وكيف يعرف التكرار المشروط ؟
- ٤٢ - ٩ هل يمكن توليد رسالة تلقين بواسطة جملة INPUT في نسخة البيسك الخاصة بك ؟ هل تلقى (لا تظهر) علامة الاستفهام ؟
- ٤٣ - ٩ هل الدوال INPUT\$, INKEY\$ متاحة على نسخة البيسك الخاصة بك ؟
- ٤٤ - ٩ هل تسمح نسخة البيسك الخاصة بك بجملة PRINT USING ؟ وإذا كان ذلك صحيحاً ، لخص الأنواع المختلفة لصيغ المخرجات المتاحة .
- ٤٥ - ٩ هل توجد جمل LPRINT, LPRINT USING على نسخة البيسك الخاصة بك ؟
- ٤٦ - ٩ ما هي الدوال المكتتبية الإضافية المتاحة في نسخة البيسك الخاصة بك ؟ وما هي الإمكانيات الإضافية المستمدة بواسطة هذه الدوال المكتتبية ؟
- ٤٧ - ٩ كيف يمكن الانتفاع من ملفات البيانات المتسلسلة في نسخة البيسك الخاصة بك ؟ كيف يمكن إنشاء ملف بيانات متسلسل ؟ وكيف يمكن إلغاؤه ؟ وكيف يمكن فتح وقفل ملفات البيانات المتسلسلة ؟ وكيف يمكن قراءة وكتابة بنود البيانات من وإلى هذه الملفات ؟
- ٤٨ - ٩ ما هو الإجراء العام لتعديل ملف بيانات متسلسل ؟ وما هي أوامر البيسك الخاصة والمتوافرة للقيام بإجراءات متعددة لملف بيانات متسلسلة
- ٤٩ - ٩ كيف يمكن الانتفاع من ملفات البيانات المشوائية في نسخة البيسك الخاصة بك ؟ كيف يمكن إنشاء وإلغاء هذه الملفات ؟ وما هو الإجراء لتعريف كل سجل ؟
- ٥٠ - ٩ كيف يمكن فتح وغلق ملفات بيانات عشوائية ؟ وكيف يمكن كتابة وقراءة بنود البيانات إلى ومن هذه الملفات ؟ هل يجب تخزين البيانات الرقمية كحروف ؟ إن كان ذلك صحيحاً ، فكيف يمكن إنجاز عملية التحويل ؟
- ٥١ - ٩ هل يتطلب مخزن الذاكرة الوسيط عند استخدام ملفات بيانات عشوائية ؟ وإن كان ذلك صحيحاً ، هل يقسم المخزن الوسيط إلى حقول ؟ وكيف يمكن إنجاز ذلك ؟

- ٥٢-٩ هل تتضمن نسخة البيسك الخاصة بك إجراءات لغة الآلة التالية :
- (أ) اختيار جزء من ذاكرة الحاسب .
 (ب) فحص محتويات ذاكرة الحاسب .
 (ج) تغيير محتويات ذاكرة الحاسب .
 (د) التوصل إلى برنامج فرعى بلغة الآلة .
- وإن كان ذلك صحيحاً ، فكيف يمكن القيام بهذه الإجراءات ؟
- ٥٣-٩ هل توجد خواص تقليدية معينة من البيسك ومستبعدة من نسخة البيسك الخاصة بك ؟ (أنظر الملاحق A و C من أجل ملخص الخواص التقليدية للبيسك) .
- ٥٤-٩ ماهي الخواص الإضافية المعدلة المتاحة في نسخة البيسك الخاصة بك ؟

مسائل للبرمجة

Programming Problems

- ٥٥-٩ أعد كتابة كل من البرامج التالية والتي تنتفع غاماً بالخواص المعدلة والمتاحة في نسخة البيسك للحاسب الدقيق الخاصة بك :
 وعلى الأخص حاول استخدام كل من IF-THEN, IF-THEN-ELSE, WHILE/WEND , PRINT USING كلما أمكن .
- (أ) جذور المعادلة التربيعية (مثال ٢ - ٣٠) .
 (ب) جذور المعادلة الجبرية (مثال ٤ - ٥) .
 (ج) حساب قيمة الاستهلاك (مثال ٤ - ٩) .
 (د) إيجاد متوسط بيانات تلوث الهواء (مثال ٤ - ١٦) .
 (هـ) إعادة ترتيب قائمة من الأرقام (مثال ٥ - ١٤) .
 (و) مولد بجلتين (مثال ٦ - ١٥) .
 (ز) لعبة صدقة أو خط (اصطيد كرايس) (مثال ٦ - ٢٠) .
 (ح) محاكاة ارتداد كرة (مثال ٦ - ٢٨) .
- ٥٦-٩ وسع برنامج مرتب الكلمات المبعثرة (مثال ٥ - ٩) بحيث يتمكن من إعادة تنظيم الحروف في أى كلمة (أى عدد من الحروف) إلى كل التوافقيات الممكنة . واستخدم كل الدوال المكتبية الموسعة المتاحة في نسخة البيسك الخاصة بك .
- ٥٧-٩ أعد كتابة البرامج المعطاة في الأمثلة ٩ - ٢٩ ، ٩ - ٣٠ وذلك لإنشاء ومعالجة مجموع النقاط التي يحرزها الطالب في الامتحانات ، وذلك باستخدام إجراءات ملف بيانات متسلسل متاح في النسخة الخاصة بك من لغة البيسك (أنظر أيضاً الأمثلة ٨ - ٢ و ٨ - ٣ و ٨ - ٤)
- ٥٨-٩ أعد كتابة برنامج تحكم المخزون المعطى في مثال ٩ - ٣١ باستخدام إجراءات ملف بيانات عشوائى متاح في النسخة الخاصة بك من لغة البيسك : (أنظر مثال ٨ - ١٠) .

- ٥٩ - ٩ أعد كتابة برنامج البحث الثنائي المعطى في مثال ٨ - ١٣ مستخدماً إجراءات ملف بيانات عشوائى متاح فى النسخة الخاصة بك من لغة البيسك .
- ٦٠ - ٩ اكتب برنامج بييسك كاملاً سوف ينشأ وينتفع من ملف بيانات متسلسل يحتوى على أسماء وعناوين وأرقام تليفونات كما ، تم توصيفها فى المسألة ٨ - ٤٠ . استخدم إجراءات ملف بيانات متسلسل متاح فى النسخة الخاصة بك من لغة البيسك .
- ٦١ - ٩ كرر المسألة ٩ - ٦٤ ولكن بالانتفاع بملف بيانات عشوائى استخدم إجراءات ملف البيانات العشوائى المتاح فى النسخة الخاصة بك من لغة البيسك . قارن ذلك مع نسخة ملف البيانات المتسلسل وذلك من وجهة نظر سهولة البرمجة وسرعة التنفيذ .
- ٦٢ - ٩ حل المسألة ٥ - ٥٧ (د) (حساب الانحراف المعياري لقائمة من الأرقام باستخدام معادلتين مختلفتين) باستخدام أرقام حقيقية (دقة مفردة ثم كرر الحسابات باستخدام دقة مزدوجة . قارن النتائج فى كل من الحالتين .
- ٦٣ - ٩ حل مسألة ٥ - ٥٥ (حساب نتائج امتحان الطلبة ، متضمناً متوسط الفصل وحياد متوسط كل طالب عن متوسط الفصل) وذلك باستخدام الخصائص المحسنة المتاحة فى النسخة الخاصة بك فى لغة البيسك للحاسب الدقيق . وعلى الأخص ، تأكد من استخدام جملة PRINT USING .

الفصل ١٠

بيئة الحاسب الدقيق .

The Microcomputer Environment

سنناقش في هذا الفصل بيئة الحاسب الدقيق من حيث المكونات المادية (Hardware) والغير مادية (Software)، أى البرامج الجاهزة . سنبدأ بمناقشة طرق تمييز الملفات ونظم أوامر البيسك للحاسب الدقيق ، ثم سنتبر الخواص الوحيدة لشاشة TV ولوحة المفاتيح للحاسب الدقيق مع عدة خصائص بيسك الحاسب الدقيق ، ثم سنتبر فيما بعد استخدام بعض أجهزة الإدخال القابلة للبرمجة واستخدام الصوت واللون ، ونختتم هذا الفصل بمناقشة مختصرة عن طرق البرامج الداخلية للتعديل .

هذا بالإضافة إلى أن اهتمامنا سيكون على استخدام ميكروسوفت بيسك ونظام التشغيل المصاحب له (MS-DOS) ، كما يتم تنفيذه على حاسب IBM الشخصى وغيره من الحاسبات الدقيقة من هذا النوع . (وتسمى شركة IBM نظام التشغيل هذا ، PC-DOS) . وعلى القارئ أن يطلع على الكتيب المرجعى المناسب للحصول على معلومات عن الأنواع الأخرى من الحاسبات الدقيقة أو نظم تشغيلها .

١٠ - ١ تمييز الملفات FILE DESIGNATIONS

يكون لأغلب الحاسبات الدقيقة جهاز تخزين ضخمة واحد على الأقل ، وعادة جهازان أو ثلاثة ، والتي يمكن تخزين أنواع مختلفة من الملفات عليها . والقرص المرن هو أكثر الأنواع شيوعاً لجهاز التخزين الضخم ، إلا أنه من المتاح أيضاً أشرطة كاسيت وأقراص صلبة وأقراص ضوئية وذاكرات فقاعية . ويمكن استخدام أى من أجهزة التخزين الكبيرة هذه لتخزين برامج البيسك ، وملفات البيانات والأنواع الأخرى من الملفات (مثل برامج لغة الآلة التى هى جزء من نظام التشغيل) . وعلى ذلك فعندما نشير إلى ملف معين تكون هناك حاجة لتحديد اسم الملف ونوع الملف وجهاز التخزين الضخم الذى تم التخزين عليه .

ويحتوى الملف الكامل المميز الموجود على حاسب IBM الشخصى (أو أى حاسب دقيق آخر يستخدم نظام تشغيل MS-DOS) على البنود الثلاثة التالية :-

- ١ - مرجع جهاز التخزين الضخم (حرف مفرد تعقبه نقطتان) .
- ٢ - اسم ملف (١ - ٨ حروف) .
- ٣ - امتداد ملف يحدد نوع الملف (١ - ٣ حروف مسبقة بنقطة) .

مثال ١٠ - ١

الآتى يبين ملف MS-DOS كاملاً مميزاً :-

A:SAMPLE.BAS

هذا الملف المميز يشير إلى برنامج يسمى SAMPLE تم تخزينه على محرك القرص A . (ومحركات الأقراص سترقم بحروف A ، B ، C) . وعلى ذلك فإن جهاز التمييز هو A ، واسم الملف هو SAMPLE والامتداد هو BAS . لاحظ النقطتين بين جهاز التمييز واسم الملف والنقطة التى تفصل اسم الملف والامتداد .

وعندما يعمل الحاسب الدقيق فإن أحد أجهزة التخزين الضخمة تكون دائماً مميزة « الجهاز النشط » (أو إذا كان محرك قرص ، فإنه يكون محركاً نشيطاً) . وعندما يتداول ملف تم تخزينه على جهاز نشيط فإنه لا حاجة إلى تحديد الجهاز المميز . ومع كل .. فعندما يتداول ملف على جهاز آخر لابد من اعتبار الجهاز المميز كجزء من توصيف الملف .

مثال ١٠ - ٢

حاسب دقيق له محركان للأقراص المرنة مميّزان كمحرك A ومحرك B. نفرض أن محرك A هو المحرك النشط حالياً، فإذا أردنا أن نشير إلى برنامج يسك على المحرك A يسمى SAMPLE فيمكننا كتابة مواصفات الملف كالآتي :-

SAMPLE.BAS

ومن ناحية أخرى إذا أردنا الإشارة إلى برنامج يسك على المحرك B يسمى DEMO، فلا بد لنا من كتابة المواصفات الكلية للملف كالآتي :-

B:DEMO.BAS

ويمكن بسهولة للملف (مثل برنامج) مخزن على أحد الأجهزة أن يتداول ملفاً آخر (مثل ملف بيانات) مخزناً على جهاز آخر . ولعمل ذلك يمكننا ببساطة جعل تمييز الجهاز كجزء من تمييز الملف عند تداول الملف الثاني .

مثال ١٠ - ٣

نفترض مرة أخرى حاسباً دقيقاً له محركان للأقراص المرنة A و B حيث المحرك A نشيط . ولنفرض إننا نقوم بتشغيل برنامج يسك مخزن على المحرك A وهذا البرنامج يتداول ملف بيانات توصل عشوائياً على المحرك B يسمى STUDENTS.DAT . وعلينا إذا الإشارة إلى ملف بيانات التوصل العشوائى كالآتي :-

B:STUDENTS.DAT

من داخل البرنامج

مثال ١٠ - ٤

شكل ١٠ - ١ يبين تغييراً في البرنامج الذى سبق كتابته في مثال ٩ - ٣٠ لتشغيل درجات امتحانات الطلبة مستخدمين معلومات مخزنة على ملفات بيانات متتابعة البرنامج الأصلي مبين في شكل ٩ - ١٠ . ومع كل فإن ملفات البيانات المتتابعة متداولة من محرك B وكل منها له امتداد DAT . متصل باسم الملف . السطور 40، 50، 250 و 260 تعكس هذه التغيرات في الاستخدام .

```

10 REM ***** PROGRAM TO PROCESS STUDENT EXAMINATION SCORES *****
20 REM ***** USING SEQUENTIAL DATA FILES *****
30 DIM C(15)
40 OPEN "I",1,"B:SCORES.DAT"
50 OPEN "O",2,"B:UPDATE.DAT"
60 INPUT #1,TITLE$
70 INPUT #1,TERM$
80 PRINT "Course title: ";TITLE$,"Term: ";TERM$
90 PRINT #2,TITLE$; PRINT #2,TERM$
100 PRINT: INPUT "Exam number";K
110 PRINT: INPUT "Calculate averages (Y/N) ";ANS$
120 PRINT: INPUT #1,N$
130 FOR I=1 TO K-1: INPUT #1,C(I): NEXT I
140 PRINT N$,
150 INPUT "New score";C(K)
160 PRINT #2,N$
170 SUM=0
180 FOR I=1 TO K: PRINT #2,C(I);: SUM=SUM+C(I): NEXT I
190 IF ANS$="N" OR ANS$="n" THEN 230
200 AVG=SUM/K
210 PRINT "Average=";AVG
220 PRINT #2,AVG
230 IF NOT EOF(1) THEN 120
240 CLOSE
250 KILL "B:SCORES.DAT"
260 NAME "B:UPDATE.DAT" AS "B:SCORES.DAT"
270 END

```

شكل ١٠ - ١

وعند تداول برنامج بيسك ، فإنه يمكن حذف كل من تمييز الجهاز وامتداد الملف بشروط معينة . وسنعرض المزيد حول خصائص ملف البيسك في القسم التالى .

١٠ - ٢ أوامر نظام الحاسب الدقيق MICROCOMPUTER SYSTEM COMMANDS

تحتوى العديد من نسخ بيسك الحاسب الدقيق على أوامر معينة للنظام والتي لا توجد فى النسخ التقليدية للغة . ومن ناحية أخرى فإن بعض أوامر نظام تقليدى معين ليست متاحة على الحاسب الدقيق ، أو يتم تنفيذها بشكل مختلف وعلى ذلك فسنعطى بعض الاعتبارات لأوامر نظام الحاسب الدقيق الأكثر شيوعاً كما يتم تنفيذها فى ميكروسوفت بيسك .

وفى بعض الحاسبات الدقيقة يظهر البيسك أوماتيكياً بمجرد تشغيلها . ومع كل فإنه من المعتاد فى الغالب أن يتداول الحاسب الدقيق نظام تشغيله بعد البدء فى التشغيل . ولكى ندخل البيسك من نظام التشغيل ، فلا بد للمستخدم أن يطبع أمر BASIC . (هناك أوامر مشابهة متاحة لتوصيل لغات أخرى أو تطبيقات حاسبات دقيقة أخرى مثل معالجة الكلمات) .
وبمجرد تداول البيسك ، فإن المستخدم سيرى رسالة مثل التى تظهر كالتالى :—

The IBM Personal Computer Basic
Version D2.10 Copyright IBM Corp. 1981, 1982, 1983
61327 Bytes free

OK

وبعد ذلك فإنه يمكن للمستخدم تحميل برنامج البيسك وتشغيله وإظهار قائمة الخ على الشاشة وذلك بإدخال أوامر النظام المناسبة ..
وسنلخص فيما يلى أوامر النظام الأكثر شيوعاً (كما يتم تنفيذها فى ميكروسوفت بيسك لحاسب IBM الشخصى) .

الأمـر	الغرض	أمثلة
AUTO	يرقم أسطر البرنامج المتتالية أوماتيكياً (والأرقام سوف تكون 10,20,30...)	AUTO AUTO 100,5
CLEAR	يجعل كل المتغيرات الرقمية تساوى صفراً ، وكل المتغيرات لسلسلة الحروف خالية .	CLEAR
CONT	يستأنف التنفيذ بعد إيقافه (مثل جملة STOP أو جملة END) .	CONT
DELETE	يحذف كتلة من السطور من برنامج موجود حالياً فى الذاكرة	DELETE 50 - 80 تحذف السطور من 50 إلى 80
EDIT	يدخل أسلوب تنقيح السطر (انظر قسم ١٠ - ٧)	EDIT 100 (لينقح السطر 100)
FILES	يعرض قائمة بكل الملفات التى تم تخزينها على وحدة تخزين كبيرة (مثل قرص مرن)	FILES FILES "B:"
KILL	يحذف ملفاً من وحدة التخزين (مثل قرص مرن)	KILL "SAMPLE" KILL "B:STUDENTS.DAT"
LIST	يعرض قائمة بكل أو بجزء من البرنامج المخزن حالياً فى الذاكرة	LIST LIST 100-200 (اعرض قائمة بالسطور من 100 إلى 200)
LLIST	يعرض (بطبع) قائمة بكل أو بجزء من البرنامج المخزن حالياً فى الذاكرة وذلك على طابعة السطور	LLIST LLIST 100-200 (اعرض قائمة بالسطور من 100 إلى 200)

الأمر الغرض أمثلة
 LOAD يسترجع ملفاً من على وحدة التخزين (مثل قرص مرن) ويخزنه في (DEMO هو اسم الملف)
 LOAD "B: SAMPLE. BAS" ذاكرة الحاسب .
 MERGE يدمج البرنامج المخزن على وحدة التخزين الكبيرة (مثل قرص مرن) مع البرنامج "TRIAL"
 MERGE "B:REPORT.BAS" الموجودة حالياً في الذاكرة .
 NAME يعيد تسمية ملف مخزن على وحدة تخزين كبيرة (مثل قرص مرن) "SAMPLE" AS "NEW PROG"
 NAME "B: DEMO" AS "B: INVADERS.BAS"
 NEW يحذف البرنامج المخزن حالياً في الذاكرة
 RENUM يعيد ترقيم سطور البرنامج أوتوماتيكياً (الأعداد ستكون 10 ، 20 ، 30 ، ...)
 RENUM 10,100,5 (إذا تم تخصيص غير ذلك) .
 (ابدأ بالسطر الأصلي 10 وغيره للسطر 100.زيادة قدرها 5)

RUN RUN ينفذ البرنامج الموجود حالياً في الذاكرة أو يحمل برنامجاً ثم يقوم بتنفيذه .
 RUN "B:SAMPLE"
 SAVE SAVE يخزن البرنامج الموجود حالياً في الذاكرة على وحدة تخزين كبيرة (مثل القرص المرن) .
 SAVE "B:SAMPLE"
 SYSTEM SYSTEM يخرج من البيسك إلى نظام التشغيل
 TRON TRON يساعد في تصحيح الخطأ ويسبب عرض أرقام سطور البرنامج عند تنفيذها .
 TROFF TROFF يلغى أمر

لاحظ أن الملفات المميزة المطلوبة محصورة بين علامات التنصيص (quotes). وهذه خاصية لميكروسوفت بييسك . ولا بد أن يكون مفهوماً أن التفاصيل التركيبية قد تختلف في النسخ الأخرى من بييسك الحاسب الدقيق . هذا بالإضافة إلى أن الأوامر نفسها قد تختلف في النسخ الأخرى من اللغة .

مثال ١٠ - ٥ تحديد الوقت من اليوم Time of Day

مثال ١٠ - ٢ يوضح جلسة نموذجية تحتوي على استخدام البيسك على حاسب IBM الشخصي . وفي هذه الجلسة نرى ماذا يحدث عندما نبدأ بتشغيل الجهاز ويتم تداول البيسك . ونبدأ بتحميل برنامج يسمى DEMO.BAS والمخزن على القرص المرن النشط (محرك A) ، ثم يتم عرض قائمة البرنامج وتشغيله . وبعد ذلك نقوم بتغيير البرنامج وذلك بإضافة جملتين PRINT إضافيتين (السطرين 35 ، 65) لعرض قائمة النسخة الجديدة ، ونقوم بتشغيلها ثم نحفظ النسخة الجديدة على المحرك النشط . وفي النهاية نخرج من البيسك ونعود إلى نظام التشغيل .

وعند البداية فإن البرنامج نفسه يقرأ اسماً مثل "Sharon" ثم يحدد الوقت الحالي من اليوم (لاحظ أنه يتم تسجيل الوقت عند بدء تشغيل الحاسب) . ويقوم البرنامج عند ذلك بطبع الجمل الملائمة مثل "Good morning, Sharon" أو "Good afternoon, Sharon" أو "Good evening, Sharon" وفي هذه الجلسة بالذات ، فإن البرنامج يطبع "Good evening, Sharon" حيث أننا أدخلنا الساعة 19:36 (أى الساعة 7.36 مساءً) عند بداية الجلسة . (لاحظ أن البرنامج يستخدم عدة دوال مكتوبة جديدة ، كما هو مبين بقسم ٩ - ٣) .

ولاحظ أن البرنامج (الملف) المميز لا يحتاج إلى تضمين مواصفات المحرك حيث أنه يتم تحميل وحفظ البرنامج في المحرك النشط حالياً . ولاحظ أيضاً أن الامتداد مفروض أن يكون BAS . وحيث أن البرنامج الحالي ليس له هذا الامتداد ، فإننا لن نحتاج إلى تضمينه في الملف المميز . وكما في الأمثلة السابقة من هذا النوع ، تكون استجابة المستخدم تحتها خط .

وتذكر أن هذا المثال هو تمثيل للجلسة برجة حاسب دقيق بميكروسوفت بييسك . وقد تختلف التفاصيل من حاسب إلى آخر . هذا بالإضافة إلى أن الجلسة قد تختلف كلية مع بعض نسخ أخرى من بييسك الحاسب الدقيق .

ولا بد للقارئ أن يرجع إلى ملحق (د) الملخص أكثر شهولاً لميكروسوفت بييسك يحتوي على أوامر النظام الأكثر استخداماً .

Current date is Tue 1-01-1980
 Enter new date: 9-5-1985
 Current time is 0:00:12.74
 Enter new time: 19:36

The IBM Personal Computer DOS
 Version 2.10 (C) Copyright IBM Corp 1981, 1982, 1983

A>BASIC

The IBM Personal Computer Basic
 Version D2.10 Copyright IBM Corp. 1981, 1982, 1983
 61327 Bytes free

Ok

LOAD "DEMO"

Ok

LIST

```
10 REM *** "Good morning" program ***
20
30 INPUT "Hi, what's your name? ",N$
40 HOUR = VAL(LEFT$(TIME$,2))
50 IF HOUR < 12 THEN PRINT "Good morning, "; ELSE IF HOUR < 18 THEN
   PRINT "Good afternoon, "; ELSE PRINT "Good evening, ";
60 PRINT N$
70 END
```

Ok

RUN

Hi, what's your name? Sharon

Good evening, Sharon

Ok

35 PRINT

65 PRINT

LIST

```
10 REM *** "Good morning" program ***
20
30 INPUT "Hi, what's your name? ",N$
35 PRINT
40 HOUR = VAL(LEFT$(TIME$,2))
50 IF HOUR < 12 THEN PRINT "Good morning, " ELSE IF HOUR < 18 THEN
   PRINT "Good afternoon, "; ELSE PRINT "Good evening, ";
60 PRINT N$
65 PRINT
70 END
```

Ok

RUN

Hi, what's your name? Sharon

Good evening, Sharon

Ok

Ok

Ok

SAVE "DEMO"

Ok

SYSTEM

A>

١٠ - ٣ شاشة عرض TV THE TV MONITOR

معظم الحاسبات الدقيقة وكثير من النهايات الطرفية للمشاركة الزمنية تستخدم شاشة عرض TV (أى وحدة CRT) كجهاز مخرجات أساسى . وتعرض هذه الوحدات فى المعتاد 24 أو 25 سطراً من النص ، وبعدد حروف يصل إلى 80 حرفاً فى السطر (ومع ذلك فيجب أن نلاحظ أن بعض الحاسبات الدقيقة تولد مخرجات ذات حروف كبيرة ، ويترتب على ذلك عدد حروف أقل فى السطر وربما أيضاً سطور أقل) . ويمكن أيضاً توليد العرض البيانى على هذه الأجهزة (انظر فصل ١٢) . وكل من نوعى شاشة العرض ذات اللون الواحد والمتعددة الألوان شائعة الاستخدام حالياً .

مسح الشاشة Clearing the Screen

ولشاشات عرض TV خصائص فريدة معينة لابد من أخذها فى الاعتبار عند كتابة برامج البيسك . فمثلاً .. يمكن لشاشة عرض TV أن تعرض عدداً محدوداً فقط من من السطور فى وقت واحد (وذلك بمقارنتها بطابع السطور الذى يمكنه أن يستمر طالما كان هناك ورق كافٍ) . وعلى ذلك فلا بد أن لمسح الشاشة دورياً حتى يمكن أن يبدأ عرض جديد فى أعلى الركن الأيسر من شاشة خالية .

وعلى حاسب IBM الشخصى تستخدم الجملة CLS (مسح الشاشة) لمسح الشاشة . وهذا الأمر يسمح كل ما كان معروضاً على الشاشة ثم يحرك المؤشر إلى أعلى الركن الأيسر .

تحريك المؤشر Moving the Cursor

توجد خاصية أخرى فريدة لشاشة عرض TV وهى المؤشر ، وهذه رمز غماز ، عبارة عن مستطيل صغير أو سطر أفقى ، ويستخدم ليبين موضعاً محدداً على الشاشة (مثل سطر أو عمود معين) . والمؤشر له عدة أغراض مفيدة ، فمثلاً يمكن استخدامه لجذب الانتباه لمكان معين على الشاشة (لقراءة رسالة مثلاً) أو طلب بيانات مدخلة أو لبيان مكان ظهور رسالة المخرجات التالية .

وعند تصحيح ملف (مثلاً برنامج البيسك) فإن موضع المؤشر تتحكم فيه لوحة المفاتيح . وتحتوى معظم لوحات مفاتيح الحاسب الدقيق على مفاتيح خاصة تستخدم لتحريك المؤشر إلى أعلى أو إلى أسفل أو على الجانبين (وستحدث عن ذلك بإفاضة فى القسم التالى) . ومع كل ، فعند تنفيذ برنامج فإن تحديد موضع المؤشر يتم داخل البرنامج . والقدرة على تحريك المؤشر بهذه الطريقة مهمة على وجه التحديد عند كتابة برامج تفاعلية (نخط تحاورى) . وعلى ذلك فإن معظم نسخ البيسك للحاسب الدقيق تحتوى على جمل خاصة لتحديد موضع المؤشر تحت تحكم البرنامج .

وعند تنفيذ برنامج ميكروسوفت بيسك على حاسب IBM الشخصى ، فإن موضع المؤشر يتم تحديده بجملة LOCATE . ويمكن أن تظهر جملة LOCATE نموذجية كالتالى :-

LOCATE 5,20

وهذه الجملة الخاصة تتسبب فى تحديد وضع المؤشر على الصف 5 (أى الصف الخامس من أعلى) والعمود 20 (من اليسار) . ويمكن ببساطة تحديد مواضع الصفوف والأعمدة الأخرى بتغيير المعالم الرقمية فى هذه الجملة .

ومن المهم أن نعرف أنه يمكن تحريك المؤشر فى أى مكان على الشاشة بهذه الطريقة بدون مسح أى جزء من النص المكتوب سابقاً . وعلى ذلك .. فإنه يمكن تحريك المؤشر إلى كلمة معينة أو رمز معين موجود فعلاً على الشاشة .

مثال ١٠ - ٦

إذا أردنا مسح الشاشة ثم وضع المؤشر فى مركز الشاشة (أى الصف 13 والعمود 40) على حاسب IBM الشخصى ، فإننا نكتب .

```
10 CLS
20 LOCATE 13,40
30 ANS$ = INPUT$(1)
```

لاحظ أن الجملة الأخيرة تسبب وقفة في تنفيذ البرنامج ، وبذلك تسمح لرؤية المؤشر في موضعه الجديد . ويستأنف تنفيذ البرنامج إذا ما تم الضغط على أى مفتاح .

ولا تحتوى بعض نسخ ميكروسوفت بيسك القديمة على أى جمل لتحريك المؤشر صراحة . وفي هذه الحالات يمكن محاكاة جملة CLS بكتابة PRINT CHR\$(12) حيث 12 هى الرقم (العشرى) لكود ASCII لتنفيذ سطر . ويمكن محاكاة جملة LOCATE أيضاً بطباعة مجموعة من السطور الخالية المتتالية ويتبعها عدة أماكن خالية في السطر المطلوب .

مثال ١٠ - ٧

حتى يمكن مسح الشاشة ووضع المؤشر في مركزها (أى صف 13 عمود 40) باستخدام حاسب دقيق من النوع القديم بدون أوامر حركة المؤشر صراحة ، يمكننا كتابة

```
10 PRINT CHR$(12)
20 FOR ROW = 1 TO 12 : PRINT : NEXT ROW
30 FOR COL = 1 TO 40 : PRINT " "; NEXT COL
40 INPUT " ",A$
```

(قارن مع مثال ١٠ - ٦)

هذه الطريقة ، بالطبع ، أقل ملاءمة من المعروضة في مثال ١٠ - ٦ . ومن ثم فيجب ألا تستخدم إلا إذا كانت هى الطريقة الوحيدة المتاحة .

توجد ثلاث دوال تستخدم مع جملة PRINT من أجل وضع المؤشر على السطر . هى (TAB) ، SPC وSPACE\$. أولها (TAB) تستخدم لوضع المؤشر في بداية السطر نسبياً كما نوقش في الفصل الخامس . والدالة الثانية (SPC) تسبب تحريك المؤشر عدداً معيناً من الفراغات وراء نطاق آخر وضع له (وهو عادة مجاور لآخر حرف مطبوع) . والدالة الأخيرة (SPACE\$) تشبه SPC حيث أنها تعيد سلسلة حروف تحتوى على عدد مميز من الأماكن الخالية .

مثال ١٠ - ٨

```
10 PRINT "RED";TAB(6);"BLUE"
```

سوف تتسبب في طباعة السلسلة الحرفية BLUE ابتداء من العمود 6، (وهو حقيقة العمود السابع في السطر) بينما الجملة :

```
20 PRINT "RED";SPC(6);"BLUE"
```

سوف تتسبب في ظهور 6 مسافات خالية بين RED, BLUE (ومن ثم فسوف تبدأ BLUE في العمود رقم 9 ، وهو حقيقة العمود العاشر في السطر) .

ويمكن كتابة جملة PRINT الأخيرة كالتالى :

```
20 PRINT "RED";SPACE$(6);"BLUE"
```

سوف تعيد دالة POS المكان الحالى للمؤشر في سطر ما . وهذا يسمح بتفرع مشروط مَقاس إلى مكان المؤشر كما هو موضح فيما بعد . (لاحظ أن هذه الدالة تهمل القيمة الموصوفة للخلاصة المطلوبة) .

مثال ١٠ - ٩

ادرس الجملتين التاليتين :

```
10 PRINT NAME$;
20 IF POS(1) < 20 THEN PRINT TAB(20); ADDRESS$
ELSE PRINT SPC(2); ADDRESS$
```

سوف تبدأ السلسلة الحرفية الثانية \$ ADDRESS في العمود رقم 20 (حقيقة العمود رقم 21) إذا كان طول سلسلة الحروف الأولى (NAME\$) أقل طولاً من 20 حرفاً ، وإلا فسوف تبدأ سلسلة الحروف الثانية بعد ثلاثة أعمدة من نهاية سلسلة الحروف الأولى (سوف يفصل السلسلتين الحرفيتين مكانان خاليان) .

اللفة الرأسية Vertical Scrolling

خاصية أخرى من خواص شاشة عرض TV هي خاصية اللفة الرأسية . وهذا يشير إلى الحركة الرأسية للكتابة المعروضة على الشاشة عند إدخال سطر جديد من النص أو عرضه على شاشة وهي ممتلئة تماماً . وعند حدوث ذلك ، فإن السطر الذى كان في أعلى الشاشة سابقاً سوف يفقد ، وكل السطور الباقية سوف تتحرك إلى أعلى بمقدار مكان سطر والسطر الجديد يظهر عند قاع الشاشة .

يمكن أن تستخدم اللفة الرأسية في بعض الأحيان على نحو مفيد لإعطاء انطباعات خاصة ، كما هو موضح في مثال ١٠ - ١٠ فيما بعد . ويمكن أيضاً أن يكون مزعجاً ، حيث أنه يمكن أن يتسبب في اختفاء نص معروض قبل إتمام قراءته ، ولكن هذه المشكلة يمكن عادة حذفها (أو تقليلها على الأقل) باستخدام الطرق الفنية الخاصة للبرمجة مثل حلقات تكرارية فارغة FOR-TO وذلك لتوليد تأخير في الوقت ، واستخدام دوال INPUT INKEY لإنشاء وقفات . انظر مثال ٩ - ٢٢ ، أو إعادة وضع المؤشر رأسياً أو استخدام آلة الطباعة لعرض النص الطويل .

ومشكلة تتعلق بذلك عن قرب هي مسح الشاشة (بواسطة جمل CLS) قبل أن تتمكن من قراءة رسالة معروضة . ويمكن أيضاً التغلب على هذه المشكلة من خلال استخدام حلقات تكرارية فارغة FOR-TO أو دوال INKEY\$ أو INPUT\$

مثال ١٠ - ١٠ برمجة شاشة عرض TV (لا شئ يمكن أن يفضى إلى خطأ ، يفضى إلى خطأ ..)

Programming a TV Display (Nothing Can Go Wrong, Go Wrong, Go Wrong,...)

نرى في شكل ١٠ - ٣ برنامج حاسب دقيق قصير للتسلية مكتوب بميكروسوفت بيسك لحاسب IBM الشخصي وهو يوضح استخدام تأخير الوقت ، وحركة المؤشر واللفة الرأسية المعتمدة . وقد صمم البرنامج ليتسبب في ملء شاشة ذات 80 حرفاً بهذه الرسالة

GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG!

لكي نملأ شاشة 80 حرف ولف رأسياً وبذلك يخلق الانطباع الكاذب بأن الحاسب الدقيق والذي تتق فيه والمفروض إلا يخطئ قد فعل ذلك حقيقة .

وهذا البرنامج يحتوى على عدد من أوامر CLS (السطور 30, 80, 200) والعديد من أوامر LOCATE (السطور 100, 130, 200, 220, 240, 30, 50, 08) .

وتسمح هذه الأوامر لمخطط البرامج أن يتحكم في مظهر الشاشة وخصوصاً مواضيع طباعة الرسائل المتعددة . ويحتوى البرنامج أيضاً على حلقتين تكراريتين فارغتين FOR-TO (السطران 70, 120) وهما اللتين تنشعان تأخير في الوقت . ويكون التأثير النهائي هو عرض ديناميكي كرسوم متحركة بتأثير مرئى قوى .

ونشجع القارئ أن يُجرى هذا البرنامج حتى يشاهد الملف الرأسى الحقيقى . وهذا يزيد كثيراً من تفهم القارئ للبرنامج . وإنه لمن المشوق أيضاً أن يقوم بتنفيذ البرنامج على آلة طباعة أو وحدة طرفية ويقارن أثرها مع عرض شاشة TV .

```

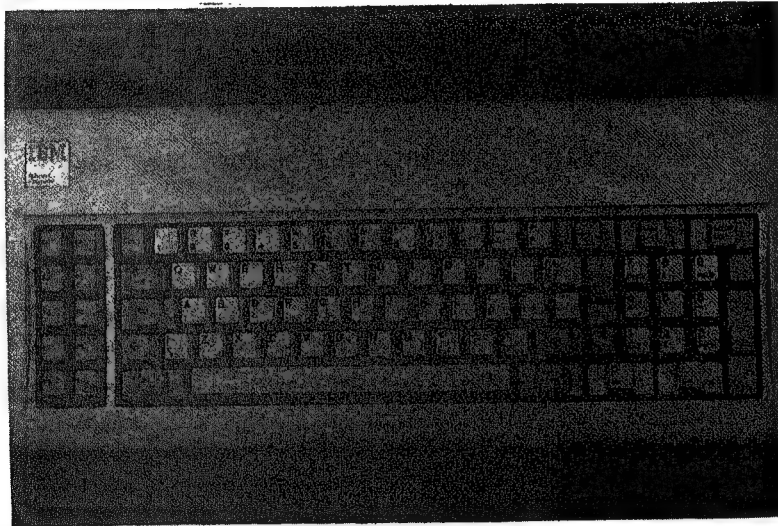
10 REM *** TV-DISPLAY DEMO (IBM Version 1) ***
20
30 CLS : LOCATE 11,28
40 PRINT "WELCOME TO THE WONDERFUL"
50 LOCATE 14,28
60 PRINT "WORLD OF MICROCOMPUTERS!"
70 FOR I=1 TO 3000 : NEXT I
80 CLS: LOCATE 11,18
90 PRINT "Just relax, enjoy yourself and remember that"
100 LOCATE 14,25
110 PRINT "NOTHING CAN POSSIBLY GO WRONG!"
120 FOR I = 1 TO 3000 : NEXT I
130 LOCATE 24
140 FOR I = 1 TO 24
150   FOR J = 1 TO 8
160     PRINT "GO WRONG! ";
170   NEXT J
180   PRINT I PRINT
190 NEXT I
200 CLS: LOCATE 11,37
210 PRINT "DARN!"
220 LOCATE 15,30
230 PRINT "Something went wrong!"
240 LOCATE 23
250 END

```

شكل ١٠ - ٣

١٠ - ٤ لوحة المفاتيح THE KEYBOARD

تحتوى معظم لوحات مفاتيح الحاسبات الدقيقة على عدد مفاتيح أكبر كثيراً من مفاتيح الآلة الكاتبة العادية. ومن المعتاد وجود مجموعتين أو ثلاث إضافية من المفاتيح. وهذه المفاتيح الإضافية قد تحتوى على مجموعة من مفاتيح المؤشر وحركة الشاشة، ولوحة مفاتيح صغيرة رقمية (أى مجموعة من المفاتيح الرقمية التى تشابه المفاتيح الرقمية للآلة الكاتبة إلا أن لها تنظيماً معيناً يشابه نوع الآلة الحاسبة) ومجموعة من مفاتيح خاصة للدالة. ويمكن برجة عدد معين من هذه المفاتيح لتقوم بأعمال خاصة خلال تنفيذ برامج بيسك.



شكل ١٠ - ٤

اعتبر مثلاً لوحة المفاتيح المستخدمة مع حاسب IBM الشخصى المبين فى شكل ١٠ - ٤ . ولاحظ أن مفاتيح الآلة الكاتبة موجودة فى مركز لوحة المفاتيح . وعلى اليمين توجد مجموعة من المفاتيح للقيام بعمليتين مختلفتين : وهى تستخدم عادة كمفاتيح للمؤشر ولحركة الشاشة . رغم أنه يمكن أن تصبح مفاتيح عددية إذا كان مفتاح « NumLock » (الموجود فى أعلى الركن الأيمن) قد تم الضغط عليه . وتحتوى الجهة اليسرى من لوحة المفاتيح على مجموعة من مفاتيح الدالة معنونة F1 حتى F10 .

وتحتوى النسخة المتقدمة من ميكروسوفت بيسك التى تصاحب حاسب IBM الشخصى (تسمى BASICA) على العديد من الأوامر الخاصة التى تسمح للحاسب بتحديد إذا ما كانت أى من مفاتيح الدالة أو مفاتيح معينة لتحريك الشاشة قد تم الضغط عليه * . يمكن بعد ذلك أن يستجيب الحاسب لأى طريقة قد يراها المبرمج من خلال استدعاء برامج فرعية خاصة (أى من خلال شكل معين من جملة GOSUB - ON) وسنرى كيف يتم ذلك فى المثال التالى .

مقال ١٠ - ١١ برمجة مفاتيح الدالة Programming the Function Keys

يمثل شكل ١٠ - ٥ برنامج بيسك بسيطاً مكتوباً بواسطة ميكروسوفت بيسك المتقدم (BASICA) لحاسب IBM الشخصى . وهذا البرنامج يحتوى على حلقة تكرارية تعرض الأرقام الصحيحة الموجبة المتتالية 1,2,3...32767 على الشاشة مع رقم واحد صحيح على كل سطر . وحيث أن الشاشة يمكنها أن تعرض 24 سطراً فى وقت واحد فإن الأرقام تتحرك رأسياً إلى أعلى بمجرد أن تمتلئ الشاشة .

والبرنامج يحتوى على امكانية انهاء الحسابات فى أى وقت يريده المستخدم ببساطة وذلك بالضغط على مفتاح الدالة F2 . كما يمكن ايضا عكس ألوان الشاشة (أى تغير الحروف البيضاء على خلفية سوداء إلى حروف سوداء على خلفية بيضاء والعكس بالعكس) فى أى وقت وذلك بالضغط على مفتاح الدالة F1 أثناء تنفيذ البرنامج .

```

10 REM *** FUNCTION KEY DEMO ***
20 '
30 ON KEY(1) GOSUB 160
40 ON KEY(2) GOSUB 200
50 KEY(1) ON : KEY(2) ON
60 '
70 CLS : FLAG = 1
80 PRINT "FUNCTION KEY DEMONSTRATION" : PRINT
90 PRINT "Press F1 to REVERSE the screen, F2 to STOP" : PRINT
100 PRINT "Press any other key to begin"
110 DUMMY$ = INPUT$(1)
120 FOR COUNT = 1 TO 32767 : PRINT COUNT : NEXT COUNT
130 END
140 '
150 REM *** SUBROUTINE TO REVERSE THE SCREEN ***
160 IF FLAG = 1 THEN COLOR 0,7 : FLAG = 0 ELSE COLOR 7,0 : FLAG = 1
170 RETURN
180 '
190 REM *** SUBROUTINE TO TERMINATE THE COMPUTATION ***
200 END

```

شكل ١٠ - ٥

دعنا الآن نختبر هذا البرنامج بالتفصيل . السطران 30,40 يصاحبان كل مفتاح دالة برنامج فرعى معين ، والسطر 50 ينشط هذه المصاحبة . السطر 70 يمسخ الشاشة ثم يخصص قيمة 1 إلى FLAG مشيراً إلى الحروف البيضاء على خلفية سوداء ، ثم تظهر بعد ذلك عبارة مبدئية فى السطر 80 حتى 10 . وتولد الأرقام الصحيحة المتتالية وتطبع فى الحلقة التكرارية FOR-TO والتى تظهر على السطر 120 ، والسطر 130 يمثل نهاية الجزء الرئيسى للبرنامج .

* فى الحقيقة يدعم حاسب IBM الشخصى ثلاثة أنواع من نسخ ميكروسوفت بيسك ، تسمى كاسيت بيسك Cassette BASIC ، ديسك بيسك Disk BASIC وبيسك المتقدم Advanced BASIC على الترتيب .

والسطور 150 إلى 170. تكون البرنامج الفرعى الأول (مصاحباً F1) والذي يعكس الألوان . والجزء الرئيسى من البرنامج الفرعى هو جملة IF-THEN-ELSE على السطر 160 . ويتم عكس الألوان بجملة COLOR . وعلى ذلك فإذا كانت FLAG=1 مشيرة إلى أبيض على أسود ، فإن الألوان تعكس بتحديد COLOR 0,7 (0 يشير إلى الحروف السوداء و 7 تشير إلى الخلفية البيضاء) . وبعد ذلك يتم وضع FLAG=0 . وبالمثل فإذا كانت FLAG لا تساوى 1 فإن الألوان تعكس بتحديد COLOR 7,0 (حروف بيضاء وخلفية سوداء) . ثم تعود FLAG إلى 1 . وجملة RETURN فى سطر 170 تسبب التحكم فى الرجوع للجملة التى تلى الموضع الذى تم فيه الضغط على مفتاح الدالة .

ويحتوى البرنامج الفرعى الثانى ببساطة على ملاحظة تحديدية وجملة END . وليس المطلوب هنا جملة RETURN حيث أن الغرض من هذا البرنامج الفرعى هو إنهاء الحسابات .

وستحدث بإفاضة عن جملة COLOR فيما بعد فى هذا الفصل (انظر قسم ١٠ - ٦) . ومع كل فلايد أن يظهر لنا أن هذه الجملة تستخدم لإنتاج نص ملون على خلفية من الألوان المختلفة، ويمكن أيضاً أن تستخدم لتوليد بيانات ملونة كما سنرى فى الفصل الثانى عشر .

وفى النهاية فإنه على القارئ أن يرى هذا البرنامج أثناء تشغيله حتى يمكنه استيعاب كيفية عمله وما يقوم به . وإذا كان حاسب IBM الشخصى متاحاً ، فإننا نشجع القارئ بشدة أن يقوم بإدخال البرنامج وتشغيله ضاغطاً على FI عدة مرات أثناء تنفيذ البرنامج ، ثم الضغط على F2 لإيقاف تنفيذ البرنامج .

١٠ - ٥ وحدات إدخال أخرى قابلة للبرمجة

OTHER PROGRAMMABLE INPUT DEVICES

يمكن للحاسبات الدقيقة أن تستخدم العديد من الأنواع المختلفة لأجهزة الإدخال القابلة للبرمجة بالإضافة إلى لوحة المفاتيح . وهناك نوعان منها يشيع استخدامهما ، وهما القلم الضوئى وعصا التوجيه . والقلم الضوئى هو جهاز للإشارة يمكن برمجته للتعرف على موقع ما على الشاشة ، ثم يقوم بتنشيط دالة ما . وبالمثل فإن عصا التوجيه هى جهاز لتحديد موضع يمكن برمجته لتحديد موضع المؤشر ثم يقوم بتنشيط دالة بالضغط على زر . ويسمح ميكروسوفت بيسك ببرمجة هذه الأجهزة بما يشابه طريقة مفاتيح دالة لوحة المفاتيح كما هو موضح فى قسم ١٠ - ٤ .

اعتبر مثلاً القلم الضوئى الذى يستخدم كما هو موضح فى شكل ١٠ - ٦ . وهو جهاز حساس للضوء وبه مفتاح فى نهايته . وهذا المفتاح يتأثر بضغط القلم الضوئى على مصدر ضوئى على شاشة عرض TV . وموضع هذا المصدر الضوئى (صف وعمود) يمكن أن يتحدد بواسطة الحاسب .

ويحتوى ميكروسوفت بيسك المتقدم على جمل خاصة متعددة ودوال صممت لتستخدم مع القلم الضوئى . واستخدامات هذه الخصائص الخاصة موضحة فى المثال التالى .



شكل ١٠ - ٦

مثال ١٠ - ١٢ برمجة قلم ضوئى Programming a Light Pen

نرى فى شكل ١٠ - ٧ برنامج بيسك متقدم مكتوب لحاسب IBM الشخصى والذى يسمح للمستخدم أن يختار واحداً من عدة بنود معروضة على شكل قائمة على شاشة عرض TV. ويتم الاختيار بواسطة قلم ضوئى. وبمجرد اختيار أحد البنود يتم اتخاذ إجراء خاص بهذا الاختيار.

وسيقوم البرنامج على وجه الخصوص بتوليد قائمة تحتوي على أسماء ثمانى لغات مختلفة، وتسبق كل منها كتلة مربعة من الضوء. ويقوم المستخدم بالضغط على طرف القلم الضوئى مقابلاً للكتلة التى تناظر اللغة التى يختارها. وبعد ذلك تظهر عبارة ترحيب باللغة المختارة بجوار اسم اللغة. وستظل عبارة الترحيب على الشاشة لمدة ثانية أو ثانيتين (وهى فترة طويلة تكفى لقراءتها بالراحة) ثم يتم مسحها. وبعد ذلك قد يختار المستخدم لغة أخرى وتم إعادة العملية. ويستمر ذلك حتى يختار المستخدم آخر سطر فى القائمة (END)والذى يتسبب فى إيقاف البرنامج.

دعنا الآن نختبر البرنامج الحقيقى ببعض التفصيل. السطر 30 يصاحب إشارة تصدر من القلم الضوئى مع برنامج فرعى. والسطر 40 ينشط هذه المصاحبة. والسطر 50 يقرأ سلسلة الحروف التى تحتوى على كتلة من الضوء. ويتم توليد القائمة بواسطة السطور 90 إلى 190. والسطر 200 يدور فى حلقة تكرارية حول نفسه عدداً غير محدود من المرات مسبباً انتظار الحاسب لإشارة من القلم الضوئى.

```

10 REM *** LIGHT PEN DEMO ***
20 '
30 ON PEN GOSUB 240
40 PEN ON
50 SQUARE#=CHR*(219)+CHR*(219)
60 '
70 REM *** MAIN LOOP ***
80 '
90 CLS : PRINT "Multi-lingual greetings"
100 LOCATE 4,1 : PRINT "Please select a language"
110 LOCATE 6,4 : PRINT SQUARE$;SPC(4);"English"
120 LOCATE 8,4 : PRINT SQUARE$;SPC(4);"French"
130 LOCATE 10,4 : PRINT SQUARE$;SPC(4);"German"
140 LOCATE 12,4 : PRINT SQUARE$;SPC(4);"Hawaiian"
150 LOCATE 14,4 : PRINT SQUARE$;SPC(4);"Hebrew"
160 LOCATE 16,4 : PRINT SQUARE$;SPC(4);"Italian"
170 LOCATE 18,4 : PRINT SQUARE$;SPC(4);"Japanese"
180 LOCATE 20,4 : PRINT SQUARE$;SPC(4);"Spanish"
190 LOCATE 22,4 : PRINT SQUARE$;SPC(4);"END"
200 GOTO 200
210 '
220 REM *** SUBROUTINE TO PRINT 'HELLO' ***
230 '
240 ROW=PEN(6)
250 IF ROW=22 THEN LOCATE 23,1 : END
260 LOCATE ROW,20
270 IF ROW=6 THEN PRINT "Hello" : GOTO 350
280 IF ROW=8 THEN PRINT "Bonjour" : GOTO 350
290 IF ROW=10 THEN PRINT "Guten tag" : GOTO 350
300 IF ROW=12 THEN PRINT "Aloha" : GOTO 350
310 IF ROW=14 THEN PRINT "Shalom" : GOTO 350
320 IF ROW=16 THEN PRINT "Buon giorno" : GOTO 350
330 IF ROW=18 THEN PRINT "Konichihua" : GOTO 350
340 IF ROW=20 THEN PRINT "Buenos dias"
350 FOR COUNT=1 TO 1000 : NEXT COUNT
360 LOCATE ROW,20 : PRINT SPACE*(15)
370 RETURN
380 END

```

شكل ١٠ - ٧

وبمجرد اكتشاف إشارة من القلم الضوئي (نتجت من القلم الضوئي بالضغط على إحدى الكتل الضوئية مسبباً إطلاق مفتاحه) فإن تحكم البرنامج يقفز أتوماتيكياً إلى السطر 240 . وهذه الجملة تستخدم دالة PEN . ويتسبب المعامل 6 في أن تعيد الدالة رقم الصف ، حيث تم تنشيط القلم الضوئي ، ثم يتم تخصيص هذه القيمة للمتغير ROW . وبعد ذلك يحرك السطر 260 المؤشر للاستعداد لعرض عبارة التحية المناسبة . والسطور 270 إلى 340 تختار عبارة التحية المناسبة وتتسبب في عرضها . ويتم توليد تأخير الوقت بواسطة الحلقة التكرارية الفارغة FOR—TO في سطر 350 . وفي النهاية يتم مسح التحية (تظهر مسافات خالية بدلاً منها) في سطر 360 ، ويعود التحكم إلى السطر 200 . وتستمر هذه العملية حتى يتم اختيار الصف 22 مسبباً إيقاف البرنامج (السطر 250) .

ويتسبب تنفيذ البرنامج في ظهور قائمة على الشاشة مشابهة لتلك المبينة في شكل ١٠ - ٨ . (لاحظ أن الكتل القائمة تظهر ككتل من الضوء على شاشة عرض TV) . ويقوم المستخدم بالضغط على القلم الضوئي مقابل إحدى الكتل لتظهر التحية المناسبة على يمين القائمة المختارة حالياً . وعلى سبيل المثال إذا ضغطنا على القلم الضوئي مقابل للكتلة الثانية ستظهر التحية "Bonjour" على يمين كلمة "French" . وستختفي التحية بعد تأخير قصير في الوقت . ويمكن بعد ذلك أن يقوم المستخدم باختيار آخر . ولاحظ أنه يمكن إنهاء البرنامج في أي وقت ببساطة وذلك باختيار الكتلة الأخيرة .

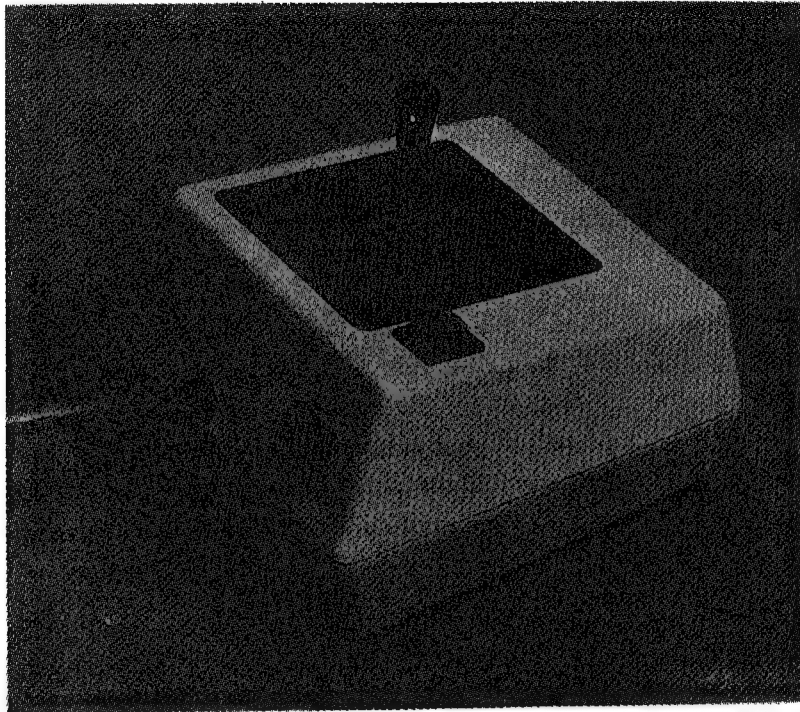
Multi-lingual greetings

Please select a language

- English
- French
- German
- Hawaiian
- Hebrew
- Italian
- Japanese
- Spanish
- END

شكل ١٠ - ٨

نفترض الآن عصا توجيه نموذجية كما هو مبين في شكل ١٠ - ٩ . ولهذا الجهاز يد (أى عصا) يمكن تحريكها في اتجاهين مختلفين (أعلى / أسفل ويسار/ يمين) . ويتم وصف موضع اليد بواسطة زوج من القيم الصحيحة تمثل الإحداثيين x و y (أى أنه يمكن تمثيل كل إحداثي بواسطة رقم صحيح ينحصر بين 0 إلى 127) . وهذه القيم ترسل للحاسب باستمرار حيث يمكن تحويلها إلى موضع للمؤشر (أى رقمى الصف والعمود) . وإذا ما تحركت يد عصا التوجيه فإن موضع المؤشر يتغير نتيجة لذلك . وعلى ذلك يمكن برمجة الحاسب لتحريك المؤشر حول الشاشة نتيجة لحركة يد عصا التوجيه .



شكل ١٠ - ٩

ويوجد لعصا التوجيه زران (مفتاحان) يمكن استخدامهما لإرسال إشارات مفردة للحاسب . وعلى ذلك فإنه يمكن أن نضع المؤشر في المكان المطلوب ، وذلك بتحريك يد عصا التوجيه ، ثم تنشيط دالة ما بالضغط على أحد الأزرار . ويمكن برمجة كل زر بمفرده لتنشيط دالته الفريدة .

ويحتوى ميكروسوفت بيسك المتقدم على العديد من الجمل والدوال الخاصة التي يمكن استخدامها لبرمجة عصا التوجيه . ويتم توضيح هذه الطريقة بالمثل التالى

مثال ١٠ - ١٣ برمجة عصا التوجيه Programming a Joystick

يحتوى شكل ١٠ - ١٠ على برنامج بيسك متقدم يسمح بتشغيل حاسب IBM الشخصى تحت تحكم عصا التوجيه . ويبدأ البرنامج بمسح الشاشة كلية وبذلك يمكن للمستخدم أن يحرك المؤشر إلى أى موضع مرعوب فيه ، وذلك بتحريك يد عصا التوجيه . ويمكن للمستخدم عند أى موضع للمؤشر أن يعرض على الشاشة نجمة (*) ، وذلك بالضغط على الزر 1 (وهو الزر الأقرب لليد فى شكل ١٠ - ٩) . ويمكن أيضاً مسح أى نجمة موجودة بالضغط على الزر 2 .

```

10 REM *** JOYSTICK DEMO ***
20 '
30 CLS
40 ON STRIG(0) GOSUB 170
50 ON STRIG(4) GOSUB 200
60 STRIG(0) ON : STRIG(4) ON
70 '
80 REM *** MAIN LOOP ***
90 '
100 X=STICK(0) : Y=STICK(1)
110 ROW=23*(Y/127) + 1
120 COL=79*(X/127) + 1
130 LOCATE ROW,COL,1
140 GOTO 100
150 '
160 REM *** SUBROUTINE TO PRINT AN ASTERISK ***
170 PRINT "*"; : RETURN
180 '
190 REM *** SUBROUTINE TO PRINT A BLANK SPACE ***
200 PRINT " "; : RETURN
210 END

```

شكل ١٠ - ١٠

ويبدأ البرنامج بمسح الشاشة (سطر 30) . والسطران 40 و 50 يصاحبان إشارات تأتى من الزرين 1 و 2 مع البرامج الفرعية التى تطبع نجمة أو مكاناً خالياً على الترتيب ، ثم يقوم السطر 60 بتنشيط هذه المصاحبة .

السطور 100 إلى 140 تعرف حلقة تكرارية وهى التى تشكل قلب البرنامج . ويكتشف السطر 100 القيم الرقمية للمتغيرين X و Y ويتوقف ذلك على موضع يد عصا التوجيه . وستقع كل من هاتين القيمتين ما بين 0 و 127 . وعلى ذلك فإن النقطة $X=0, Y=0$ تشير إلى أعلى الركن الأيسر . والنقطة $X=127, Y=127$ تشير إلى أسفل الركن الأيمن ، ثم يقوم السطران 110 و 120 بتحويل هذه القيم إلى رقم الصف (وتنحصر من 1 إلى 24) ورقم العمود (وينحصر من 1 إلى 80) . ويقوم السطر 130 بتحريك المؤشر إلى هذا الموقع الجديد . وفى النهاية ينهى السطر 140 الحلقة التكرارية (ويكشف بذلك قيمة جديدة لكل من X و Y استجابة لحركة يد عصا التوجيه) ويعيد تحديد وضع المؤشر حسب ذلك .

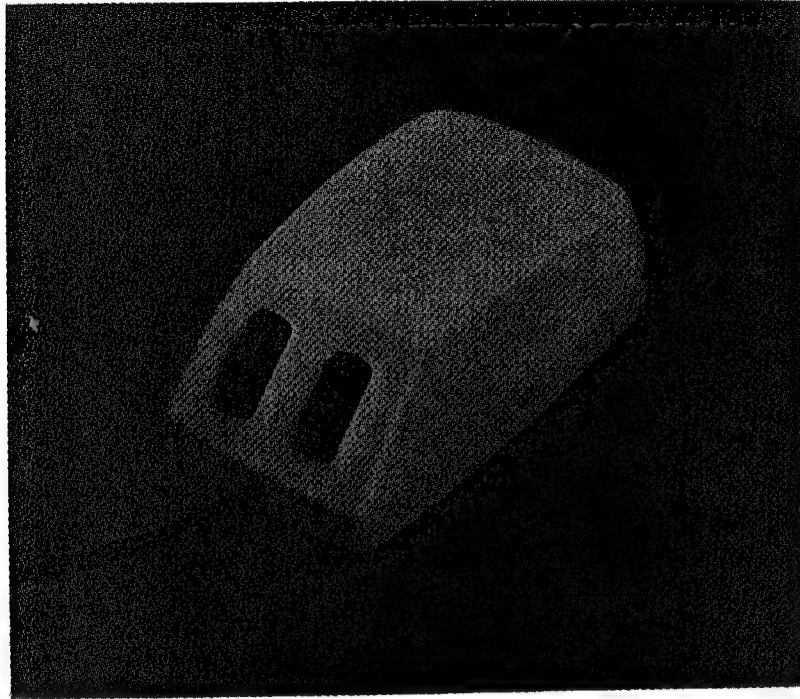
ويعرف السطر 170 برنامجاً فرعياً بسيطاً يطبع نجمة (*) في موضع المؤشر الحالي . وهذا البرنامج الفرعى يتم تنشيطه حينما نضغط على الزر 1 لعصا التوجيه أثناء تنفيذ الحلقة التكرارية الرئيسية . وبالمثل فإن السطر 200 يعرف البرنامج الفرعى الذى يطبع مسافة فارغة عند موضع المؤشر الحالي (وبذلك يتم مسح أى شئ آخر موجود عند هذا الموضع) . ويتم تنشيط هذا البرنامج الفرعى بالضغط على الزر 2 أثناء تنفيذ الحلقة التكرارية الرئيسية .

لاحظ أن هذا البرنامج يستمر فى التنفيذ بغير حدود ، أو إلى أن يخرج الحاسب من البيسك بواسطة بعض أوامر من المستخدم . ويمكن تحسين البرنامج بإضافة برنامج فرعى ينهى الحسابات عندما نضغط على مفتاح دالة كفا فى مثال ١٠ — ١١ .

وبالرغم من أن القلم الضوئى وعصا التوجيه من وحدات الإدخال الشائعة للحاسب الدقيق ، إلا أنه لا يمكن اعتبارهما بأى حال الأجهزة الوحيدة المتاحة ، فالحاسبات الدقيقة الحديثة تدعم أجهزة التحويل الرقمية وأجهزة التعرف على الكلام والعديد من الأنواع الأخرى لأجهزة الإدخال .

ومن الأمور ذات الأهمية الخاصة تزايد الاهتمام باستخدام جهاز يعرف بالمتجول (Mouse) . وهو جهاز لتحديد موضع مشابهاً لعصا التوجيه . وهذا المتجول يسهل التحكم فيه ، وذلك بتحريك المؤشر بواسطة انزلاق المتجول على سطح أملس عريض (عبارة عن مساحة حوالى قدم مربع واحد) بدلاً من عمل تصحيحات دقيقة ليد عصا التوجيه الصغيرة والحساسة .

الشكل ١٠ — ١١ يبين صورة فوتوغرافية لمتجول نموذجى . لاحظ أن هناك زرَّين عند مقدمة المتجول (وبعض هذه المتجولات لها ثلاثة أزرار ، وأخرى لها زر واحد فقط) . وكما يحدث فى عصا التوجيه فإن المتجول يسمح بتنشيط مختلف الدوال فى أى وقت وذلك ببساطة بالضغط على أحد هذه الأزرار .



شكل ١٠ — ١١

وقد أصبح استخدام المتجول شائعاً في تطبيقات ميكنة بعض المكاتب خصوصاً لمعالجة الكلمات والتطبيقات التي تستخدم بكثرة البيانات . وتدعم بعض نسخ يسك الحاسب الدقيق استخدام المتجول (ومع مجموعة من الأوامر تماثل تلك المستخدمة مع عصا التوجيه) والأخرى لا تقوم بذلك . ومع كل فتضمن أوامر المتجول في يسك الحاسب الدقيق لابد أن يزداد شيوعاً في السنوات القادمة .

١٠ - ٦ استخدام اللون والصوت USE OF COLOR AND SOUND

تدعم بعض الحاسبات الدقيقة استخدام شاشات عرض TV الملونة ، فتسمح لنص بلون واحد أن يعرض على خلفية بلون مختلف . ويستخدم عرض النص الملون في إعطاء شكل مؤثر لتحسين البرنامج ، وعلى الأخص إذا كان تفاعل البرنامج عالياً . ولتلك الحاسبات الدقيقة التي تدعم استخدام اللون يسمح اليسك باختيار الألوان بواسطة جملة COLOR . وهذه الجملة تحتوي عادة على ثلاثة معالم ، تحدد لون الواجهة الأمامي (النص) ، ولون الخلفية ولون الحافة (أى المساحة الخلفية على طول الحدود الخارجية للشاشة) . وعلى سبيل المثال .. فعلى حاسب IBM الشخصي يمكن اختيار الألوان الخلفية كالآتي

0 black	8 gray
1 blue	9 light blue
2 green	10 light green
3 cyan	11 light cyan
4 red	12 light red
5 magenta	13 light magenta
6 brown	14 yellow
7 white	15 high intensity white

ويمكن اختيار أى لون من الألوان الثمانية الأولى (0 إلى 7) للخلفية أو للحافة . وعلى ذلك فإن جملة COLOR 7,0,0 تدل على نص أبيض على خلفية سوداء وحافة سوداء ، بينما COLOR 0,7,7 تختار نصاً أسود على خلفية بيضاء وحافة بيضاء ، Color 14,1,4 تبين نصاً أصفر على خلفية زرقاء بحافة حمراء وهكذا . ويمكن تعديل قيمة المعالم في أى وقت أثناء تنفيذ البرنامج ، ويتسبب هذا في تغيير الألوان حسب المطلوب . وعادة ما يسبق استخدام جملة COLOR جملة SCREEN والتي تحدد الأسلوب (النص مقابل البيانات) وحالة اللون (ممكن أو غير ممكن) . وعلى ذلك فعلى حاسب IBM الشخصي تحدد جملة SCREEN 0,1 أسلوب النص مع إمكان اللون .

مثال ١٠ - ١٤ النص المتعدد الألوان Multicolored Text

لتوضيح استخدام نص ملون على حاسب IBM الشخصي نعرض برنامج كتب بميكروسوفت يسك يعرض على الشاشة نفس السطر من النص مكرراً ، ولكن بمعالم ألوان مختلفة . وينتج عن هذا لف رأسى على الشاشة مع عرض كل سطر بلون مختلف . وبداية فإن .. كل من الخلفية والحافة تكون سوداء . ومع كل .. فإن البرنامج يحتوى على وسيلة لتغيير ألوان الخلفية والحافة في أى وقت ببساطة وذلك بالضغط على مفتاح الدالة F1 . ويستخدم أيضاً مفتاح الدالة F2 لإيقاف الحسابات .

ويبين شكل ١٠ - ١٢ البرنامج كاملاً . ويبدأ بالأمر SCREEN فى السطر 30 الذى يحدد أسلوب النص مع إمكان اللون . والسطر 40 يحذف العرض على الشاشة لأى تعاريف سابقة لمفتاح دالة فى السطر الأسفل للشاشة . وتولد السطور 50 إلى 70 رسالة فى أسفل الشاشة والتي تصف الاستخدام الحالى لمفاتيح الدالة أى :

F1: Change background color F2: End

وتحتاج هذه المجموعة الأخيرة من الأوامر إلى بعض التفسيرات الإضافية . وعلى حاسب IBM الشخصي فإن السطر الأسفل المعروض على الشاشة (أى السطر 25) هو سطر لا يلف ويستخدم عادة لعرض تلقينات ورسائل من هذا النوع . وعادة ما تعرض مثل هذه الرسائل باللون معكوسة (نص أسود على خلفية بيضاء) وذلك لتمييزها عن أى شيء آخر قد يظهر على الشاشة . وعلى ذلك يقوم السطر 50 بوضع المؤشر في أسفل الركن الأيسر من الشاشة ويحدد لوناً عكسياً . ويولد السطر 60 الرسالة الحقيقية على السطر الأسفل ، ويحتفظ السطر 70 باللون المعتاد (نص أبيض على خلفية سوداء) ثم يعيد وضع المؤشر أعلى الركن الأيسر للشاشة .

```

10 REM *** COLOR TEXT DEMO ***
20 '
30 SCREEN 0,1
40 KEY OFF
50 LOCATE 25,1 : COLOR 0,7
60 PRINT "F1: Change background color      F2: End ";
70 COLOR 7,0 : LOCATE 1,1
80 ON KEY(1) GOSUB 210
90 ON KEY(2) GOSUB 250
100 KEY(1) ON : KEY(2) ON
110 FOREGROUND=1 : BACKGROUND=0
120 '
130 REM *** MAIN LOOP ***
140 '
150 COLOR FOREGROUND,BACKGROUND,BACKGROUND
160 PRINT "Programming with BASIC is lots of fun! "
170 FOREGROUND=(FOREGROUND+1) MOD 16
180 GOTO 150
190 '
200 REM *** SUBROUTINE TO CHANGE BACKGROUND COLOR ***
210 BACKGROUND=(BACKGROUND+1) MOD 8
220 RETURN
230 '
240 REM *** SUBROUTINE TO TERMINATE THE COMPUTATION ***
250 COLOR 7,0,0
260 END

```

شكل ١٠ - ١٢

وتحدد السطور 80 إلى 100 استخدام مفاتيح الدالة كما هو مبين في مثال ١٠ - ١١ ، ثم تتحدد خطة اللون المبدئ (نص أزرق على خلفية سوداء) في سطر 110 .

ويتم توليد لف النص بحلقة تكرارية تتكون من السطور 180 . وكل مرور خلال الحلقة التكرارية يولد سطرًا من النص ، وكل سطر في النص سيتم عرضه بلون مختلف . وعلى ذلك فالسطر 150 يختار الألوان الحالية حسب القيم المخصصة لمعالم FOREGROUND (للون النص) و BACKGROUND (للخلفية والحافة) . ويولد السطر 160 السطر الحقيقي من النص ، أى

Programming with BASIC is lots of fun!

ويولد السطر 170 قيمة جديدة للمعلم FOREGROUND للإعداد للمرور التالى خلال الحلقة التكرارية. (لاحظ أن FOREGROUND تأخذ قيمةً صحيحةً متتابةً بين 0 و 15 وذلك لاستخدام المعامل MOD). وفى النهاية يقلل السطر 180 الحلقة التكرارية ، وذلك بالعودة للسطر 150 . ويحدد السطران 210 و 220 البرنامج الفرعى الذى يتم تنشيطه بفتح الدالة F1 . ويسبب هذا البرنامج الفرعى زيادة قيمة BACKGROUND بمقدار 1 كلما ضغطنا على F1 . (لاحظ أن BACKGROUND تأخذ قيمةً صحيحةً متتابةً بين 0 و 7) . وهذا يسبب بدوره تغيير لون الخلفية والحافة بمجرد عودة التحكم إلى الحلقة التكرارية الرئيسية .

وبالمثل يعرف السطران 250 و 260 البرنامج الفرعى الذى يتم تنشيطه بواسطة مفتاح الدالة F2 . وهذا البرنامج الفرعى يقوم ببساطة بالاحتفاظ بنظام اللون القياس (نص أبيض ، خلفية سوداء ، حافة سوداء) وينهى الحسابات .

وبمجرد البدء في التنفيذ يبقى البرنامج في الحلقة التكرارية الأساسية بدون تحديد إلى أن يتم الضغط على مفتاح الدالة F2 ويظهر كل سطر من النص بلون مختلف ، إلا أن الألوان تكرر نفسها كل 16 سطراً حيث يوجد 16 فقط من الألوان الأمامية المختلفة . وسيكون هناك سطر واحد في كل مجموعة من 16 غير قابل للقراءة ، وذلك لأن اللون الأمامي يكون مائلاً للألوان الخلفية . ويتغير لون الخلفية كل مرة يتم فيها الضغط على F1 . وتبدأ سلسلة الألوان الخلفية في التكرار بعد ثمانية ألوان مختلفة حيث أن هناك ثمانية ألوان خلفية متاحة فقط .

وعلى القارئ أن يشغل البرنامج على نوع من حاسب IBM مجهزاً بشاشة ملونة إذا أمكن ذلك . وهذا يعطى للقارئ تفهماً أكبر للتأثير الذى ينشأ .

وإذا كان الحاسب مجهزاً بشاشة عرض TV ذات لون واحد (أى أسود وأبيض أو أسود وأخضر) بدلاً من شاشة عرض ملونة فإن جملة COLOR يمكن تفسيرها تفسيراً مختلفاً نوعاً ما . وعلى الأخص فإنه قد يتم استخدام معالم معينة لتحديد خواص النص مثل وضع خط تحته ، كثافة عالية أو نص ومضى وسرى مثلاً لذلك في مثال ١٠ - ١٧ .

ونعيد انتباهنا الآن لاستخدام الصوت في برنامج يسك فمعظم الحاسبات الدقيقة تحتوى على بوق داخلي يمكن تنشيطه تحت تحكم البرنامج . رغم أن هذه الأبواق تكون صغيرة إلا أنها يمكنها إصدار أنواع مختلفة ملحوظة من الأصوات . ويمكن تحسين العديد من برامج اليسك عن طريق استخدام مثل هذه البرامج التى تولد الأصوات .

وعلى سبيل المثال توجد جملة BEEP على حاسب IBM الشخصى والتي تنبه ببساطة البوق لأحداث صوت قصير ذى نغمة عالية بتردد ثابت . وتفيد مثل هذه التنبيهات بلفت انتباه المستخدم لبعض الأحداث المعينة التى يمكن أن تحدث أثناء تنفيذ البرنامج . وعادة ما يستخدم هذا التنبيه فيما يتصل ببرامج الأخطاء (انظر فصل ١١) . وهناك فوائد أخرى لها أيضاً .

مثال ١٠ - ١٥

دعنا نعدل برنامج ميكروسوفت يسك الموضح في المثال السابق بحيث يقوم البوق بالتنبيه عندما يتغير لون الخلفية (أى عندما يتم الضغط على مفتاح الدالة F1) . وهذا يمكن أن يتم بسهولة جداً ، وذلك بإضافة جملة BEEP للبرنامج الفرعى بالسطور 210 إلى 220 . وعلى ذلك فإن البرنامج الفرعى يمكن كتابته كالآتي

```
200 REM *** SUBROUTINE TO CHANGE BACKGROUND COLOR ***
210 BACKGROUND=(BACKGROUND+1) MOD 8
215 BEEP
220 RETURN
```

(لاحظ إضافة سطر 215) . ولابد للقارئ أن يجرى هذا التغيير وأن ينفذ هذا البرنامج إذا كان ذلك ممكناً بأى حال للتعرف على التأثير الذى ينشأ .

ويحتوى أيضاً ميكروسوفت يسك على جملة SOUND ، والتي تستخدم لإيجاد صوت فترته وتردده متغيرين . وكلما كان التردد عالياً ، كانت نغمة الصوت عالية . وبالمثل كلما طالت الفترة طال الصوت . وعلى سبيل المثال فإن جملة SOUND 5000,1 تنتج صوتاً قصيراً ذا نغمة عالية نسبياً ، بينما SOUND 100,2000 تولد نغمة طويلة ومنخفضة (تمثل الملعمة الأولى التردد ، والثانية تمثل الفترة) .

ويمكن بتجريب عدد معين من جملة SOUND يتضح ذلك كثيراً حيث أنه يمكن الحصول على عدة أصوات مشوقة باستخدام هذه الجملة . ويكون هذا صحيحاً على الأخص إذا كانت الجملة تدخل ضمن الحلقة التكرارية FOR—TO بقيم متغيرة للمعالم . وبعض هذه التأثيرات موضحة في المثالين التاليين .

مثال ١٠ - ١٦ برمجة البوق (A Siren) Programming the Speaker

يمثل شكل ١٠ - ١٣ برنامج ميكروسوفت يسك مكتوب لحاسب IBM الشخصى . وهو يبين كيف يولد البوق الداخلى صوت السرينة . ويحتوى البرنامج أساساً على الحلقة التكرارية FOR—TO التى تتكرر عدداً غير محدود من المرات . وفى كل مرة يتم فيها تنفيذ الحلقة التكرارية ينتج عنها صراخ واحد من السرينة . وهذا الصوت المخصوص فى السرينة يتكرر كل مرة يتم فيها تنفيذ الحلقة التكرارية بالكامل . وتستمر السرينة بغير حدود حتى يتم الضغط على مفتاح الدالة F1 .

دعنا نختبر البرنامج بالتفصيل . فالسطر 30 بمسح عرض أى تعاريف سابقة لمفتاح الدالة من أسفل الشاشة ، ويولد السطران 40 و50 الرسالة

Press F1 to STOP

في مركز الشاشة . والسطر 60 يصاحب مفتاح الدالة F1 برنامج فرعى لانتهاء الحسابات والسطر 70 ينشط هذه المصاحبة ويتحدد طول كل صوت منفرد بتخصيص القيمة 0.02 إلى DURATION في السطر 80. (لاحظ أن كل صوت منفرد يكون قصيراً) .

والسطر 120 إلى 150 تنتج صوتاً حقيقياً للسريّة . وفي السطور 120 إلى 140 نرى الحلقة التكرارية FOR—TO التي تنتج صراخاً واحداً للسريّة وكل مرور خلال هذه الحلقة التكرارية يولد نغمة قصيرة يتحدد ترددها بالدليل (FREQUENCY) . لاحظ أن النغمة سيزداد ترددها أثناء المرور كل مرة خلال الحلقة التكرارية . وحيث أن النغمات تكون قصيرة ، فإن التأثير يكون كما في حالة إيجاد صوت سريّة يرتفع باستمرار بالرغم من أن الأنغام المفردة لها ترددات غير مستمرة ومتصلة وفي النهاية يجعل السطر 150 الحلقة التكرارية تبدأ من جديد مولدة بذلك تكرار صوت السريّة .

```

10 REM *** SOUND DEMO (SIREN) ***
20 '
30 KEY OFF
40 CLS : LOCATE 12,32
50 PRINT "Press F1 to STOP";
60 ON KEY(1) GOSUB 180
70 KEY(1) ON
80 DURATION=.02
90 '
100 REM *** MAIN LOOP ***
110 '
120 FOR FREQUENCY=400 TO 1000 STEP 5
130   SOUND FREQUENCY,DURATION
140 NEXT FREQUENCY
150 GOTO 120
160 '
170 REM *** SUBROUTINE TO TERMINATE THE COMPUTATION ***
180 END

```

شكل ١٠ - ١٣

وتستمر العملية بالكامل حتى يتم الضغط على مفتاح الدالة F1 . وهذا يقوم بتحويل التحكم في الحال إلى جملة END في السطر 180 مسبباً إيقاف البرنامج .

ومرة أخرى فإننا نشجع القارئ على تشغيل هذا البرنامج فعلاً حتى يمكنه أن يفهم ماذا يحدث .

والمثال التالي يوضح الاستخدام المركب لـ COLOR ، BEEP و SOUND لتحسين برنامج ببسك الذي يولد عرضاً بسيطاً لنص . وهذا البرنامج يتم تنفيذه على شاشة عرض TV بلون واحد . وعلى ذلك فإن جملة COLOR تستخدم لاختيار خواص مختلفة للنص مثل الكثافة العالية أو الوضعية بدلاً من الألوان الحقيقية .

مثال ١٠ - ١٧ برمجة شاشة عرض TV Programming a TV Display

شكل ١٠ - ١٤ يبين تعديراً في البرنامج الأصلي الذي سبق عرضه في مثال ١٠ - ١٠ الذي يتسبب في ظهور الرسالة

GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG!

بملاً شاشة من 80 حرف والتي تلف رأسياً . ويكون المنطق هو نفسه أساساً كما في البرنامج الذي تم عرضه سابقاً (أنظر شكل ١٠ - ٣) . ومع كل .. فهذه النسخة من البرنامج يتم تحسينها بإضافات عديدة من جملة COLOR والعديد من جملة BEEP و SOUND فمثلاً .. تسبب جملة COLOR 15,0 في السطر 40 الرسالة الافتتاحية .

**WELCOME TO THE WONDERFUL
WORLD OF MICROCOMPUTERS!**

التي يتم عرضها بكثافة عالية والتي تعطى تأكيداً خاصاً على هذه الرسالة . وبالمثل فإن جملة COLOR 15,0 في السطر 230 تسبب الكلمة

DARN!

في الظهور بكثافة عالية معطية ثانياً تأكيداً خاصاً . (لاحظ أن النص ذا الكثافة العالية يختفى بواسطة جملة COLOR 7,0 في السطرين 90 و 280) .

ويمكن الحصول على تأثير مختلف نوعاً ما بجملة COLOR 16,7 في السطر 150 . وهذه الجملة تسبب عرض السطور التي تحتوي على الرسالة .

GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG! GO WRONG!

وذلك كحروف سوداء ومضيه على خلفية بيضاء (أى على شكل عرض ومضى معكوس) . وعلى ذلك يضيف لمسة درامية أثناء لف السطور إلى أعلى الشاشة .

ويمكن تحسين البرنامج باستخدام جملتي BEEP في السطرين 40 و 90 وجمل SOUND في السطرين 210 و 260 . وتسبب جملتنا BEEP فيام البوق بالتنبيه عندما يبدأ عرض افتتاحية كل رسالة ملفتاً الانتباه لهذه الرسالة . وبالمثل فإن جملتي SOUND في السطر 210 تخلفان نغمة منخفضة وتنبيهاً أطول مصاحباً لكل سطر من سطور النص التي تلف (وتقوم أول جملة SOUND بتوليد الصوت الحقيقي) . كما تولد جملة SOUND الثانية وقفة (أى صوت بتردد مرتفع جداً لا يمكن سماعه) . وبدون هذه الوقفة فإن الأصوات التي تنتج بواسطة جملة SOUND الأولى ستستمر معاً أثناء المرور المتتابع في الحلقة التكرارية FOR—TO محدثة بذلك نغمة واحدة طويلة مستمرة .

```

10 REM *** TV-DISPLAY DEMO (IBM Version 2) ***
20
30 KEY OFF
40 CLS : COLOR 15,0 : LOCATE 11,28 : BEEP
50 PRINT "WELCOME TO THE WONDERFUL"
60 LOCATE 14,28
70 PRINT "WORLD OF MICROCOMPUTERS!"
80 FOR I=1 TO 3000 : NEXT I
90 CLS : COLOR 7,0 : LOCATE 11,18 : BEEP
100 PRINT "Just relax, enjoy yourself and remember that"
110 LOCATE 14,25
120 PRINT "NOTHING CAN POSSIBLY GO WRONG!"
130 FOR I = 1 TO 3000 : NEXT I
140 LOCATE 24
150 COLOR 16,7
160 FOR I = 1 TO 16
170     FOR J = 1 TO 8
180         PRINT "GO WRONG! ";
190     NEXT J
200     PRINT
210     SOUND 500,5 : SOUND 32767,5 : PRINT
220 NEXT I
230 COLOR 15,0
240 CLS : LOCATE 11,37
250 PRINT "DARN!"
260 FOR I=1 TO 18 : SOUND 50,1 : SOUND 32767,1 : NEXT
270 LOCATE 15,30
280 COLOR 7,0 : PRINT "Something went wrong!"
290 LOCATE 23
300 END

```

ونرى في السطر 260 جملة SOUND متابعتين كما في السطر 210 والتي تضمنتها الحلقة التكرارية FOR — TO هو تنفيذ جملة SOUND هذه مرة بعد الأخرى . ويكون التأثير العام هو عبارة عن سلسلة من الأصوات القصيرة ذات النغمة المنخفضة وتشبه نوعاً ما نقيق الضفدعة . وصوت هذا النقيق يقصد به تحسين كلمة DARNI التي تتولد في السطر السابق . ومرة أخرى نشجع القارئ على تنفيذ هذا البرنامج فعلاً حتى يمكنه أن يفهم بالكامل التأثيرات البصرية والسمعية التي تنتج .

١٠ - ٧ تنقيح البرنامج PROGRAM EDITING

معظم نسخ ميكروسوفت بيسك تسمح بمرونة أكبر في تنقيح البرنامج أكثر عن النسخ التقليدية من اللغة . فمثلاً في ميكروسوفت بيسك يمكن أن يتم تعديل برنامج بيسك بثلاث طرق مختلفة باستخدام جملة EDIT مع خاصية تعديل الشاشة الكاملة ومع المنقح التقليدي للنص أو معالج الكلمات .

وتكون جملة EDIT أكثر ملاءمة لإجراء تصحيحات للسطور المفردة في أجزاء مختلفة من البرنامج . ولاستخدام هذه الجملة يمكن للمبرمج ببساطة كتابة EDIT ويتبعها رقم السطر ، أى

EDIT 50

ويتسبب هذا في عرض السطر على شاشة عرض TV ويكون المؤشر في وضعه عند أول السطر . ويمكن إدخال دالة التنقيح المطلوبة (أى حركة المؤشر ادخال ، حذف ... الخ) بالضغط على المفتاح المناسب .

ويمكن للقراء المهتمين الرجوع إلى الكتيب المرجعي للمبرمج فيما يخص حساباتهم الخاصة للحصول على معلومات أكثر عن هذه الخاصية المفيدة .

مثال ١٠ - ١٨

افترض في برنامج البيسك الموجود في مثال ١٠ - ١٧ (شكل ١٠ - ١٤) أن السطر 150 قد تم إدخاله بطريقة الخطأ مثل

150 COLLR 16,7

(لاحظ أن COLLR قد أدخلت بدلاً من COLOR) . ولإجراء التعديلات اللازمة يقوم المبرمج بكتابة

EDIT 150

وهذا يتسبب في ظهور السطر الأصلي (الغير صحيح) على شاشة عرض TV مع ظهور المؤشر تحت I في 150 أى

150 COLLR 16,7

وبعد ذلك يتم تحريك المؤشر إلى الحرف الغير صحيح أى

160 COLLR 16,7

ثم يتم استبدال O بدلاً من L . وبمجرد أن يتم التغيير يقوم المبرمج بالضغط على مفتاح RETURN . وهكذا يستبدل السطر الأصلي بالسطر المنقح .

وإذا كان من الضروري تغيير عدة سطور متتالية ، فقد يكون من الأنسب طباعة قائمة لكثلة من السطور كاملة على الشاشة بدلاً من تنقيح كل سطر منفرداً ، حيث يمكن القيام بالتغيير المطلوب وذلك بتحريك المؤشر إلى المكان المناسب في كل سطر ثم نعيد الطباعة حسب الضرورة . ويتسبب الضغط على مفتاح RETURN بعد كل تغيير في أن يستبدل السطر الأصلي بالسطر المنقح حديثاً . وعلى ذلك فإن المؤشر يأخذ الوضع في أول السطر التالى أوتوماتيكياً . ويشار إلى هذا بتنقيح الشاشة بالكامل .

مثال ١٠ - ١٩

نعتبر مرة أخرى برنامج البيسك المعروض في مثال ١٠ - ١٧ . ولنفرض أننا نعرف أن السطور من 160 إلى 220 تحتوى على عدة أخطاء ، فإن إحدى الطرق لتصحيح هذه الأخطاء أن نصحح كل سطر على حدة كما في المثال السابق . وقد يكون الأسهل كتابة

LIST 160-220

```

160 FOR I=1 TO 16
170   FOR J=1 TO 16
180     PRINT "GO WRING! ";
190     NEXT I
200     PRNT
210     SOUND 500 : SOUND 32767 : PRINT
220   NEXT J

```

وينتج عن هذا عرض الآتي على شاشة عرض TV (لاحظ أن كل سطر يحتوى على خطأ واحد على الأقل) وتصحيح كتلة الجمل هذه نحرك المؤشر أولاً إلى المكان المناسب في السطر 160 ونصحح الخطأ (FOR تكون FUR) ثم نضغط على مفتاح RETURN . وهذا يحرك المؤشر أوتوماتيكياً إلى بداية السطر 170 . نحرك المؤشر ونصحح الخطأ في السطر 170 ونضغط على مفتاح RETURN ، ثم نصصح الأخطاء في السطر 180 وهكذا حتى تتم جميع التغييرات . وفي نهاية هذه العملية ، ستظهر السطور من 160 إلى 220 كالآتي :

```

160 FOR I=1 TO 16
170   FOR J=1 TO 8
180     PRINT "GO WRONG! ";
190     NEXT J
200     PRINT
210     SOUND 500,5 : SOUND 32767,5 : PRINT
220   NEXT I

```

لاحظ أنه لم تكن هناك ضرورة لكتابة EDIT 160 ,*(EDIT) ... الخ لكل سطر من السطور المطلوب لتتقحها ويمكن أيضاً تنقيح برنامج ييسك باستخدام منقح نص تقليدي ، أو معالج الكلمات بشرط أن يكون مخزناً كما في ملف ASCLL بدلا من أن يكون مكتوفاً أو مضغوطة وحتى تحتفظ ببرنامج بهذه الطريقة عندما تستخدم ميكروسوفت ييسك ، فعلى المستخدم أن يطبع .

SAVE "program name",A

ويمكن أن يستدعى البرنامج إذا بواسطة منقح النص ، ويتم التغيير الضروري .

مثال ١٠ - ٢٠

نفرض أن برنامج "GO WRONG" المعروض في مثال ١٠ - ١٧ قد تم إدخاله على الحاسب لأول مرة ، ولدينا شك في وجود عدة أخطاء مطبعية في كل مكان بالبرنامج . وهذه الأخطاء يمكن تصحيحها باستخدام أى من الطريقتين الموضحتين في المثالين السابقين . ومع كل فقد يكون من الأسهل حفظ البرنامج كملف ASCII ثم إجراء التصحيحات باستخدام منقح النص .

SAVE "B:DEMO",A

ولتخزين البرنامج كملف ASCII على محرك قرص "B" ، وإجراء التصحيحات الضرورية . وبمجرد إتمام كل التصحيحات يمكن تخزين الملف الجديد مرة أخرى على المحرك B باسم DEMO. BAS وتنفيذه فيما بعد .

أسئلة للمراجعة Review Questions

- ١ - ١٠ حدد نوع نظام التشغيل المتاح على الحاسب الدقيق الموجود في منزلك أو مكتبك . حدد أيضاً كيفية تمييز الملفات مع نظام التشغيل هذا .
- ٢ - ١٠ ما هو الفرق بين اسم ملف وامتداد ملف ؟ ما هو الغرض المفيد الذي يؤديه امتداد ملف ؟.
- ٣ - ١٠ كيف يمكن لبرنامج مخزن على أحد أجهزة التخزين الكبيرة أن يتداول ملف (أى ملف بيانات) والمخزن على جهاز تخزين كبير آخر ؟
- ٤ - ١٠ أشرح الغرض من كل من أوامر نظام الحاسب الدقيق الآتية :-

AUTO, CLEAR, CONT, DELETE, EDIT, FILES, KILL, LIST, LLIST, LOAD, MERGE,
NAME, NEW, RENUM, RUN, SAVE, SYSTEM, TRON, TROFF.

- ٥ - ١٠ ما هي أوامر النظام المتاحة على الحاسب الدقيق التي تستخدمه في منزلك أو مكتبك ؟
قارن كل من هذه الأوامر مع تلك الموجودة في سؤال ١٠ - ٤ .
- ٦ - ١٠ كم من النصوص (ما عدد الصفوف والأعمدة) يمكن عرضها على شاشة عرض TV لحاسب دقيق نموذجي ؟
- ٧ - ١٠ كيف يمكن مسح الشاشة أثناء تنفيذ برنامج بيسك ؟
- ٨ - ١٠ كيف يمكن تحريك المؤشر بينما يتم تنقيح تصحيح البرنامج ؟ وكيف يمكن تحريكه أثناء تنفيذ البرنامج ؟
- ٩ - ١٠ هل يمكن تحريك المؤشر إلى موضع على الشاشة المعروف فعلاً عليها نص أثناء تنفيذ برنامج بيسك ؟ وإذا تم ذلك فهل هذا يؤدي ذلك إلى إتلاف النص الذي سبق عرضه ؟
- ١٠ - ١٠ ما هو الفرق بين دوال البيسك SPACES, SPC, TAB . متى يكون أفضل استخدام لكل داله من هذه الدوال ؟
- ١١ - ١٠ ما معنى اللف الرأسى ؟ وكيف يمكن التحكم فيه ؟
- ١٢ - ١٠ لماذا يكون تأخير الوقت مفيداً في برنامج بيسك للحاسب الدقيق ؟ وكيف يمكن توليد مثل هذا التأخير للوقت ؟
- ١٣ - ١٠ قارن بين استخدام تأخيرات الوقت المحدودة (أى تأخيرات الوقت التي تدوم لفترة معينة من الوقت) مع عدم تحديد وقت للتأخيرات (الانقطاعات التي يبقى تأثيرها حتى يضغط المستخدم على مفتاح ، الخ) . تحت أى ظرف يكون كل نوع من تأخير الوقت أكثر مناسبة ؟ .
- ١٤ - ١٠ أشرح المجموعات الرئيسية للمفاتيح على لوحة مفاتيح لحاسب دقيق معين . ما الغرض من كل مجموعة رئيسية ؟
- ١٥ - ١٠ هل تتضمن نسخة البيسك المتاحة على الحاسب الدقيق الخاص بك امكانيه برجه مفاتيح الداله ؟ وإذا كان كذلك فكيف يمكن أن يتم هذا ؟
- ١٦ - ١٠ ما هو القلم الضوئى ؟ ولأى نوع من التطبيقات تكون هذه الوسيله أكثر فائدة ؟
- ١٧ - ١٠ هل تتضمن نسخة البيسك المتاحة على الحاسب الدقيق الخاص بك امكانيه برجه قلم ضوئى ؟ وإذا كانت كذلك فكيف يمكن أن يتم هذا ؟
- ١٨ - ١٠ ما هي عصا التوجيه ؟ ولأى نوع من التطبيقات تكون هذه الوسيله أكثر فائدة ؟
- ١٩ - ١٠ هل تتضمن نسخة البيسك المتاحة على الحاسب الدقيق الخاص بك امكانية برجه عصا توجيه ؟ وإذا كان كذلك فكيف يمكن أن يتم هذا ؟
- ٢٠ - ١٠ ما هو المتجول ؟ ولأى نوع من التطبيقات تكون هذه الوسيله أكثر فائدة ؟ (قارن مع عصا التوجيه) .

- ١٠ - ٢١ هل تتضمن نسخه اليبسك المتاحه على الحاسب الدقيق الخاص بك امكانيه برجه المتجول ؟ وإذا كان كذلك فكيف يمكن ان يتم هذا ؟
- ١٠ - ٢٢ ما هو الغرض من جمله COLOR في وسيله النصوص ؟ (بالمقارنه بوسيلة البيانات) ؟ ولأى حاله تستخدم كل معلمه ؟
- ١٠ - ٢٣ كيف تفسر جمله COLOR بواسطة الحاسب الدقيق المجهز بشاشه عرض TV ذات اللون الواحد ؟ .
- ١٠ - ٢٤ ما هو الغرض من جمله SCREEN ؟ وفيه تستخدم كل معلمه عندما تكون في وسيله النصوص ؟
- ١٠ - ٢٥ هل يحتوي الحاسب الدقيق الخاص بك على جملي SCREEN, COLOR ؟ وإذا لم يكن كذلك فهل توجد جمل أخرى تؤدي نفس الشيء ؟
- ١٠ - ٢٦ ما الغرض من جمله BEEP ؟
- ١٠ - ٢٧ ما الغرض من جمله SOUND ؟ وما هو الغرض من كل معلمه ؟
- ١٠ - ٢٨ كيف يمكن استخدام جمل SOUND, BEER, COLOR لتحسين برنامج يبسك ذي طبيعه غير بيانيه ؟
- ١٠ - ٢٩ أشرح استخدام أمر EDIT . لأى نوع من حالات التنقيح يكون هذا الأمر أكثر مناسبة ؟
- ١٠ - ٣٠ أشرح كيف يمكن أن يتم التنقيح باستخدام خاصية تنقيح الشاشه بالكامل الموجوده في كثير من نسخ يبسك الحاسب الدقيق . تحت أى ظروف يكون تنقيح الشاشه بالكامل وفضلا عن تكرار استخدام أمر EDIT ؟
- ١٠ - ٣١ هل نسخه اليبسك للحاسب الدقيق الخاص بك تحتوي على أمر EDIT (أو امر مقارن) ؟ وهل يدعم ذلك تنقيح الشاشه بالكامل ؟
- ١٠ - ٣٢ قارن استخدام منقح النصوص مع استخدام محاصيه تنقيح الشاشه بالكامل والمتاحه في كثير من نسخ يبسك الحاسب الدقيق . أيهما أفضل وتحت أى ظروف ؟
- ١٠ - ٣٣ كيف يمكن حفظ برنامج يبسك على أنه ملف ASCII على نسختك الخاصه من يبسك الحاسب الدقيق ؟

أسئله تكميليه

Supplementary Problems

- المسائل الاتيه تهتم بجمع المعلومات بدلا من الحلول الحقيقيه للمسائل . أجب على الامثله كما تطبقها على نسخه اليبسك للحاسب الدقيق الخاص بك .
- ١٠ - ٣٤ كيف يمكن تمييز ملف كامل مكتوب للحاسب الدقيق الخاص بك ؟ وكيف يمكن تمييز محركات اقراص متعدده من واحد الى آخر ؟
- ١٠ - ٣٥ ما هي أوامر النظام المتاحه في نسختك من اليبسك ؟ كيف تختلف أوامر النظام هذه عن أوامر النظم التقليديه المشروحه في فصل ٣ ؟ .
- ١٠ - ٣٦ ما هو أقصى عدد من السطور يمكن عرضه على شاشه عرض TV الخاصه بك وما هو أقصى عدد للحروف لكل سطر ؟
- ١٠ - ٣٧ كيف يمكن مسح الشاشه من خلال نسختك من اليبسك ؟
- ١٠ - ٣٨ كيف يمكن تحريك المؤشر أثناء تنقيح البرنامج ؟ كيف يمكن تحريك المؤشر من خلال نسختك من اليبسك ؟
- ١٠ - ٣٩ كيف يمكن أن يبدأ اللف الرأسى ويتم التحكم فيه من خلال نسختك من اليبسك ؟
- ١٠ - ٤٠ كيف يمكن توليد ايقاف الوقت مؤقتا (أى الايقاف لفترة وقت محدود) أثناء تنفيذ برنامج مستخدماً نسختك من اليبسك ؟ كيف يمكن توليد ايقاف غير محدد الوقت (أى ايقاف لفترة وقت غير محده) ؟
- ١٠ - ٤١ كيف يمكن تحويل المخرجات من شاشه عرض TV لالة طباعة ؟ هل يتضمن هذا مخرجات مصاغه ؟

- ٤٢ - ١٠ هل تحتوي لوحه مفاتيح حاسبك الدقيق على مفاتيح حركه المؤشر و / أو مفاتيح الداله ؟ إذا كان كذلك هل توجد جمل خاصه تسمح ببرمجة هذه المفاتيح ؟
- ٤٣ - ١٠ هل حاسبك الدقيق يمدك باستخدام وسائل مساعده خاصه مثل القلم الضوئى أو عصا التوجيه أو المتجول ؟ إذا كان كذلك هل هناك جمل ييسر خاصه تسمح ببرمجة هذه الوسائل ؟
- ٤٤ - ١٠ هل حاسبك الدقيق يمدك بالنصوص الملونه ؟ إذا كان كذلك كيف يحدد اللون فى اليبسك ؟ هل يمكن تحديد لون خلفيه منفصل ؟ هل هناك لون حافه منفصل ؟
- ٤٥ - ١٠ هل هناك أوامر خاصه لتنبية البوق فى نسختك من اليبسك ؟
- ٤٦ - ١٠ هل نسختك من اليبسك تسمح بتوليد اصوات مختلفه تحت تحكم البرنامج ؟ إذا كانت كذلك ما هى الأوامر التى تستخدم لتنفيذ ذلك ؟ كيف يمكن تحديد التردد والفرقه ؟
- ٤٧ - ١٠ هل نسختك من اليبسك تحوى امكانية تنقيح سطر ؟ إذا كانت كذلك فكيف يمكن تحريك المؤشر على السطر الذى يتم تنقيحه ؟ كيف يمكن ادخال الحروف وحذفها ؟

استله للبرمجه

Programming Problems

معظم المسائل المعطاه فيما يلى تحتاج الى استخدام اللون «Color» إذا لم يكن عندك شاشه ملونه يمكن تبديل الألوان المختلفه بالخواص المختلفه لأحاديه اللون (مثل شدة ضوء عاليه أو عكس أو نص ومضى) لألوان مختلفه .

- ٤٨ - ١٠ عدل البرنامج المعطى فى مثال ١٠ - ١٤ ليحتوى على استخدام مفاتيح الدوال F1 حتى F4 . استخدام مفاتيح الدوال بالطرق الآتيه :-

- F1 يختار لون الواجهه
- F2 يختار لون الخلفيه
- F3 يختار لون الحافه
- F4 ينهى الحسابات

اجعل الحاسب ينبه «Beep» عندما يتغير اللون ربما تريد أن تبدل رساله برساله من اختيارك .

- ٤٩ - ١٠ عدل برنامج التحكم فى المخزون (مثال ٩ - ٣١) بحيث يظهر المخزون صفرا بلون مختلف (أى أحمر) عن النصوص الأخرى المعروضه على الشاشه .

- ٥٠ - ١٠ عدل برنامج توليد اعداد فيبوناشى والبحث عن الأرقام الأوليه (مثال ٩ - ١١) بحيث تظهر الأرقام الأوليه بلون مختلف عن باقى النص . أجعل الحاسب ينبه «Beep» عندما يتم طباعة عدد أولى .

- ٥١ - ١٠ السؤال ٩ - ٦٣ يطلب منك حل المسأله ٥ - ٥٥ (حساب درجات امتحانات طلبه ، متضمنا متوسط الفصل والانحراف المتوسط لكل طالب عن متوسط الفصل) باستخدام الخصائص المحسنه المتاحة فى نسختك من بيسك الحاسب الدقيق . حل المسأله ٩ - ٦٣ بإضافه استخدام اللون للبيانات الخارجيه المعروضه . وأعرض على الأخص كل النص بنفس اللون ما عدا متوسط الفصل والانحراف المتوسط لكل طالب عن متوسط الفصل . وأعرض متوسط الفصل بلون آخر والانحراف بلون ثالث .

- ٥٢ - ١٠ المسأله ٩ - ٥٥ تطلب منك إعادة كتابه مولد بجلاتين والموجود بمثال ٩ - ٢٨ بحيث يمكنها أن تتضمن عدة خصائص اضافيه مثل نص متعدد السطور ، علامات التنصيص وأصوات ثنائية الحروف .

أعد كتابه مولد بجلاتين كما هو مبين فى مسأله ٩ - ٥٥ ، بالاضافه الى ذلك أضف امكانية عرض النصوص الانجليزيه (أى البيانات المدخله) على شاشه عرض TV بلون واحد وعرض بجلاتين (المخرجات المناظره) بلون آخر . استخدم ثلاث مفاتيح دوال مختلفه لتغيير الوان النص الانجليزي ، ونص بجلاتين والخلفيه على الترتيب .

أجعل الحاسب ينبه «Beep» في بداية كل ترجمة بجلاتين (كل كتلة مخرجات جديدة) استخدم مفتاح داله لتحويل التنبيه لتشغيلها أو إبطالها .

١٠ - ٥٣ أعد كتابة البرنامج الخاص بحساب الاستهلاك (مثال ٤ - ٩) بحيث يظهر كل عمود من الاعداد على شاشة عرض TV بلون مختلف . استخدم الخواص المتقدمة التي تعطيكها نسختك من بيسك الحاسب الدقيق حيثما أمكن ذلك عمليا . واستخدم على الأخص مفاتيح الداله لاختيار الطريقة التي تستخدم لحساب الاستهلاك (أى استخدم F1 لاختيار طريقته الخط المستقيم للاستهلاك و F2 لاختيار طريقة توازن الانحراف المزدوج للاستهلاك ، F3 لاختيار طريقة مجموع أرقام السنوات و F4 لانهاء الحسابات) . اعرض قائمه مختصره عند أعلى الشاشة لتبين الغرض في كل مفتاح داله . بين نتائج كل من الحسابات على شاشته منفصله (أى امسح الشاشة بين الحسابات) .

١٠ - ٥٤ عدل برنامج «Hello» المبين في مثال ١٠ - ١٢ . (برجه قلم ضوئى) ليحتوى على استخدام لون . أعرض على الأخص قائمه اللغات بلون واحد والتحيه المناظره بآخر . اجعل الحاسب ينبه «Beep» عند عرض التحيه .

١٠ - ٥٥ أعد كتابه البرنامج المعطى في مثال ١٠ - ١٢ بحيث يمكن اختيار لغه باستخدام مفاتيح الداله بدلا من قلم ضوئى . اجعل البرنامج يحتوى على استخدام لون كما هو موجود في مثال ١٠ - ٥٤ .

١٠ - ٥٦ أعد كتابه البرنامج المعطى في مثال ١٠ - ١٢ بحيث يمكن اختيار لغه مع عصا توجيه بدلا من قلم ضوئى . اجعل البرنامج متضمنا استخدام لون كما هو مبين في مثال ١٠ - ٥٤ .

١٠ - ٥٧ أعد كتابه البرنامج المعطى في مثال ١٠ - ١٢ بحيث يمكن اختيار لغه مع المتجول بدلا من قلم ضوئى . أجعل البرنامج يحتوى على استخدام لون كما هو مبين في مثال ١٠ - ٥٤ .

١٠ - ٥٨ حل المسألة ٥ - ٥٦ (اختيار العواصم المقابله لدولها) باستخدام مفاتيح الداله لاختيار أما الدوله أو العاصمه . تأكد من استخدام الخصائص المحسنه الموجوده في نسختك من بيسك الحاسبات الدقيق عندمل يمكن ذلك عمليا . استخدم ايضا اللون والصوت كما هو موضح في مثال ١٠ - ٥٤ .

١٠ - ٥٩ كرر المسألة ١٠ - ٥٨ مستخدما كل من أجهزة الإدخال المبرجه الآتيه لاختيار أما دوله أو عاصمه .

(أ) قلم ضوئى (ب) عصا توجيه (ج) متجول

تأكد من تحسين برنامجك بالاستخدام الفعال للصوت واللون .

١٠ - ٦٠ عدل البرنامج الموجود في مثال ١٠ - ١٣ (برجه عصا التوجيه) بحيث يمكن استخدام مفاتيح الداله لاختيار الوان مختلفه للواجه والخلفيه . صمم البرنامج بطريقه بحيث يمكن عرض الوان النجوم المختلفه على خلفيه منتظمه . اسمح بإمكان تغير لون الخلفيه في أى وقت .

١٠ - ٦١ أعد مسأله ١٠ - ٦٠ باستخدام متجول بدلا من عصا توجيه .

١٠ - ٦٢ عدل البرنامج المعطى في مثال ١٠ - ١٦ بحيث يتم توليد صوت سرينه « على ومنخفض » . اجعله يحتوى على ما يمكنه تغير الوان الواجه والخلفيه اتوماتيكيا كل مره يتولد فيها صوت السرينه .

١٠ - ٦٣ اكتب برنامج بيسك والذي يولد مقياس موسيقى في مفتاح C . ضمن البرنامج ما يمكنه اختيار مقياس تصاعدى أو مقياس تناولى أو كليهما (أى المقياس الذى يصعد ثم يهبط) .

اسمح للاختيار بأن يتم بواسطه اختيار مفتاح داله مناسب . أعرض قائمه في أعلى الجزء الأيسر من الشاشة لتبين الغرض من كل مفتاح داله متاح (ويمكن أن يفيدك كتيب مرجع مخطط البرامج عن الترددات اللازم تحديدها حتى يمكن توليد النغمات الموسيقيه .

١٠ - ٦٤ اكتب برنامج بيسك يمكنه أن يلعب اعنيه بسيطه على بوق حاسبك (ويمكن لكتيب مرجع مخطط البرامج أن ان يبين كيف يمكن توليد نغمات موسيقيه بلغه بيسك لحاسبك المعين) .

الفصل ١١

البرمجة سهلة الاستخدام .

User-Friendly Programming

تشجع بيئة الحاسب الدقيق تفاعلاً على استخدام حوار المستخدم في كثير من برامج البيسك . وعادة ما تحتوي هذه الحوارات على صورة من صور تفاعل السؤال الجواب حيث يسأل الحاسب الأسئلة ويقوم المستخدم بتقديم الأجوبة . وتعمل هذه الطريقة بشكل حسن جداً عندما تكون الأسئلة بسيطة وقليلة العدد ، إلا أنه في بعض التطبيقات قد تكون الأسئلة عديدة أو معقدة أو مكررة مما ينتج عنه تشويش أو إرباك أو إحباط من جانب المستخدم . وفي ظل هذه الظروف يكثر احتمال حدوث أخطاء في إدخال البيانات . هذا بالإضافة إلى أن المستخدم قد يستخلص أن البرنامج صعب أو مثير للأعصاب بدرجة لا يستطيع معها استخدامه بسهولة وعلى ذلك فقد يتجنب استخدامه في المستقبل .

ويمكننا غالباً تجنب مثل هذه المشاكل بإدخال خصائص الاستخدامات السهلة داخل البرنامج مثل جعل التلقين والقوائم واختيارات الخطأ وتحقق المستخدم . وتسهل هذه الخصائص عامة استخدام البرنامج مع أن تبنيها عادة ما يحتاج إلى جهد كبير من جانب المبرمج . سنناقش استخدام العديد من خصائص الاستخدامات السهلة في هذا الفصل .

١١ - ١ التلقينات PROMPTS

التلقين هو أمر مختصر أو شرح يولده الحاسب قبل طلب المعلومة . وعلى ذلك قد يشرح التلقين نوع المعلومة المطلوبة أو الإجابة المسموح بها . ويتم خلق الحوار بين المستخدم والحاسب من خلال الاستخدام الذكي لمثل هذه التلقينات . وقد قابلنا بالفعل استخدام التلقينات وقد قابلنا بالفعل استخدام التلقينات في كثير من الأمثلة الموجودة في الفصول السابقة من هذا الكتاب . ومع كل فإن معظمها كان مختصراً جداً وسنعيد الآن استخدام تلقينات أكثر تفصيلاً مقترنة ببرامج تصحيح الأخطاء التي تختبر صحة البيانات المدخلة .

مثال ١١ - ١

نفرض أنه مطلوب من برنامج البيسك للحاسب الدقيق تلقين المستخدم درجات اختبارات الطلبة والتي تتراوح قيمتها بين 0 و 100 في المائة . وقد يحتوي البرنامج على جملة INPUT التالية والتي تحتوي على تلقين مقبول .

```
100 INPUT "Exam score (0-100): ",SCORE
```

ويخبر التلقين المستخدم عن المعلومات المطلوبة وتبين مدى القيم المسموح بها. ومع كل .. فالبرنامج لن يقوم باختبار تحديد ما إذا كانت القيمة التي أدخلها المستخدم فعلاً (أى القيمة التي خصصت إلى SCORE) تقع دال المدى المطلوب .

وهنا تلقين مشابه مصحوب باختبار الخطأ للقيمة العددية الغير صحيحة المدخلة .

```
100 LOCATE 3,1: INPUT "Exam score (0-100): ",SCORE
110 IF SCORE < 0 OR SCORE > 100 THEN BEEP: LOCATE 3,1:
PRINT SPACE$(30): GOTO 100
```

إذا خصص الآن لـ SCORE قيمة عددية خارج المدى المسموح به سوف يبين الحاسب الخطأ وتحذف القيمة السابقة من على الشاشة (في الحقيقة السطر كله سوف يستبدل بـ 30 مسافة فارغة) وسوف يظهر التلقين ثانياً في مكانه الأصلي . ويستمر البرنامج في الدوران في هذه الحلقة التكرارية إلى أن يدخل المستخدم قيمة مقبولة SCORE .

الجملةتان المبيتان سابقاً مناسبان لتصيد أى قيم عددية غير صحيحة ، إلا أنه تبقى إمكانية قيام المستخدم بإدخال حرف آخر غير عددى بدلاً من عدد . وهنا يكون التلقين مصحوباً باختبار الخطأ الأكثر شمولاً والذي يحتوى على اختبار للحروف غير المرغوب فيها كما يحتوى على اختبار للأرقام غير المناسبة .

```
100 LOCATE 3,1: INPUT "Exam score (0-100): ",ANS#
110 IF LEFT$(ANS#,1)="0" THEN SCORE=0: GOTO 130
    ELSE SCORE=VAL(ANS#)
120 IF SCORE <= 0 OR SCORE > 100 THEN BEEP: LOCATE 3,1:
    PRINT SPACE$(30): GOTO 100
130 . . . next statement . . .
```

الآن تكون استجابة المستخدم هى الإدخال المبدئى كسلسلة حروف (ANS\$) بدلاً من قيمة عددية . وإذا ما قام المستخدم بإدخال القيمة 0 فإن البرنامج سوف يتعرف عليه فى السطر 110 ويخصص القيمة 0 لـ (SCORE) ثم ينتقل خارج الحلقة التكرارية إلى السطر 130 . ومع كل فإذا ما أدخل أى شيء آخر فإن قيمة (ANS\$) تتحول مباشرة إلى قيمة عددية وتخصص لـ (SCORE) . وسوف تتحول أى قيمة غير عددية أوتوماتيكياً إلى 0 ، وتعتبر قيمة غير مقبولة بسبب الاختبار الموجود فى السطر 120 . وعلى ذلك فإنه يمكننا اكتشاف الحروف غير الرقمية والغیر مرغوب فيها وكذلك القيم العددية التى لاتقع خلال المدى المسموح به .

مثال ١١ — ٢ إدخال درجات امتحان طلبة **Entering Student Examination Scores**

فى مثال ٩ — ٢٩ قابلنا برنامج اليبسك الذى ينشئ ملف بيانات متتابعاً ، ويحتوى على مجموعة من درجات امتحان الطلبة . والبرنامج الأصيل مكتوب بلغة ميكروسوفت بيسك لحاسب IBM الشخصى المبين فى شكل ٩ — ٨ .

```
10 '***** CREATE A SEQUENTIAL DATA FILE *****
20 '
30 '***** (ENTER STUDENT EXAMINATION SCORES - VERSION 2) *****
40 '
50 DIM SCORE(12)
60 OPEN "D",1,"SCORES:"
70 KEY OFF: CLS: WIDTH 80
80 LOCATE 1,1: INPUT "Course title: ",TITLE#
90 LOCATE 3,1: INPUT "Term: ",TERM#
100 LOCATE 5,1: INPUT "How many exam scores per student? (1-12) ",ANS#
110 N=VAL(ANS#)
120 IF N < 1 OR N > 12 THEN BEEP: LOCATE 5,1: PRINT SPACE$(60): GOTO 100
130 PRINT #1,TITLE#
140 PRINT #1,TERM#
150 COUNT=1
160 '
170 '*** BEGIN LOOP ***
180 '
190 CLS: PRINT "Student number",COUNT
200 LOCATE 3,1: INPUT "Student name (Type END to end data entry): ",N#
210 IF N#="END" OR N#="end" THEN 330
220 PRINT #1,N#
230 FOR I=1 TO N
240     LOCATE I+4,1: PRINT "Exam number";I;" (0-100): "; INPUT "",ANS#
250     IF LEFT$(ANS#,1)="0" THEN SCORE(I)=0: GOTO 270 ELSE SCORE(I)=VAL(ANS#)
260     IF SCORE(I) <= 0 OR SCORE(I) > 100 THEN BEEP: LOCATE I+4,1:
        PRINT SPACE$(60): GOTO 240 ELSE PRINT #1,SCORE(I);
270 NEXT I
280 PRINT #1,""
290 COUNT=COUNT+1: GOTO 190
300 '
310 '*** END LOOP ***
320 '
330 CLOSE
340 END
```

دعنا الآن نغير هذا البرنامج لنجعله أكثر تحاوراً وعلى ذلك يكون أكثر سهولة في الاستخدام وذلك بإضافة بعض التلقينات الإضافية مع اختبارات الأخطاء المناظرة . ويظهر البرنامج الجديد في شكل ١١ - ١ . ومنطق البرنامج الكلي يشابه ذلك الذي استخدم في البرنامج السابق . ومع كل.. فقد أضفنا التلقينات واختبارات الخطأ المناظرة لكل من الكميات العددية المدخلة. ونلاحظ على الأخص أن السطور من 100 إلى 120 هي تلقينات لعدد درجات الامتحان لكل طالب وعندئذ نختبر القيم التي أدخلت . وبالمثل فإن السطور من 240 إلى 260 تلقن كل درجة امتحان ثم نختبر القيم التي أدخلت . وتشابه طرق اختبارات إلى حد كبير النموذج المعروض في مثال ١١ - ١ .

وبالإضافة إلى ذلك فإن هذا البرنامج يستخدم عدداً من خصائص التحكم في الشاشة وهو أكثر عمومية ، بمعنى أنه يمكن أن يقوم بتسكين من 1-12 امتحان لكل طالب . لاحظ استخدام الحلقة التكرارية FOR-TO (السطور من 230 إلى 270) لإدخال كل درجات امتحان الطالب من لوحة المفاتيح وكتابتها في ملف البيانات المتتابع .

شكل ١١ - ٢ ، ١١ - ٣ يوضحان حواراً نموذجياً لمستخدم يحدث عند تنفيذ البرنامج . ويبين شكل ١١ - ٢ الحوار المبدئي الذي يلحق عنوان المقرر والفصل الدراسي وعدد الامتحانات لكل طالب . استجابات المستخدم تحتها خط .

Course title: Comp Sci 141

Term: Fall 1985

How many exam scores per student? (1-12) 5

شكل ١١ - ٢

Student number 1

Student name (Type END to end data entry): Adams B F

Exam number 1	(0-100):	<u>45</u>
Exam number 2	(0-100):	<u>80</u>
Exam number 3	(0-100):	<u>80</u>
Exam number 4	(0-100):	<u>95</u>
Exam number 5	(0-100):	<u>55</u>

شكل ١١ - ٣

يبين شكل ١١ - ٣ الحوار الذي يطلب اسم ودرجات الامتحان لأول طالب . وهنا أيضاً تكون استجابات المستخدم تحتها خط . ونفس هذا الحوار سوف يتكرر لكل طالب تال حتى ندخل كلمة END في مكان اسم الطالب .

ومن سوء الحظ أن هذه الأشكال لا يمكنها توضيح آثار اختبارات الخطأ . وعلى ذلك فإن القارئ مطالب بتشغيل هذا البرنامج فعلاً وإدخال معلومات غير صحيحة حتى يحصل على بعض الإدراك من حيث طريقة عمل اختبارات الخطأ .

١١ - ٢ القوائم MENUS

تحتاج بعض البرامج التي يختارها المستخدم إلى أحد الاختيارات العديدة والتي قد تكون متاحة . وعادة ما يكون أفضل طريقة للاختيار هي الاختيار من القائمة ، أي عرض قائمة بجميع الاختيارات على الشاشة مع رقم أو حرف متميز مناظر لكل اختيار. ويمكن للمستخدم أن يقوم بالاختيار بطريقة مبسطة وذلك بكتابة الحرف أو الرقم المناسب . وعملية الاختيار من القائمة هذه سهلة جداً وتؤدي إلى تقليل احتمال حدوث الأخطاء .

مثال ١١ - ٣

نفترض أن برنامج ميكروسوفت بيسك المكتوب لحاسب IBM الشخصي يحتوي على إمكان اختيار واحد من أربعة ألوان بخلفيات مختلفة .
قد يحتوي البرنامج على الجمل الآتية والتي تولد قائمة وتسمح للمستخدم أن يختار منها .

```
100 CLS
110 LOCATE 1,1: PRINT "Background Colors:"
120 LOCATE 3,1: PRINT " 1 - Blue"
130 LOCATE 4,1: PRINT " 2 - Green"
140 LOCATE 5,1: PRINT " 3 - Red"
150 LOCATE 6,1: PRINT " 4 - White"
160 LOCATE 8,1: INPUT "Please enter your selection: ",ANS$
170 CHOICE=VAL(ANS$)
180 IF CHOICE < 1 OR CHOICE > 4 THEN BEEP: LOCATE 8,1:
    PRINT SPACE$(60): GOTO 160
190 ON CHOICE GOTO 200,300,400,500
```

وعند تنفيذ البرنامج فإن القائمة الآتية سوف تظهر على الشاشة .

Background Colors:

- 1 - Blue
- 2 - Green
- 3 - Red
- 4 - White

Please enter your selection:

سوف يستجيب المستخدم بإدخال 1 أو 2 أو 3 أو 4 . وبناء على ذلك يقوم البرنامج باختيار لون الخلفية الصحيح وذلك بالتفرع للسطور 200 أو 300 أو 400 أو 500 كما توجهه جملة ON—GOTO في سطر 190 .

لاحظ أن السطور من 160 إلى 180 تحتوي تلقيناً واختياراً للخطأ المناظر الذي يتسبب عندما يتم إدخال أية قيمة غير 1 أو 2 أو 3 أو 4 (بما في ذلك الحروف غير العددية) في أن ترفض . وعلى ذلك فإنه يسمح للمستخدم بأن يستجيب فقط بإدخال واحد من الاختيارات المعطاة في القائمة .

وفي هذا المثال بالذات كان يمكننا استخدام الحروف بدلاً من الأرقام للاختيار من القائمة، ويكون الاختيار المنطقي هو استخدام الحرف الأول من كل لون. وعلى ذلك فإن القائمة قد تظهر كما يأتي :-

Background Colors:

- B - Blue
- G - Green
- R - Red
- W - White

Please enter your selection: -

ويكون هذا مناسباً لهذه المجموعة الخاصة من الألوان ، حيث أن كل لون يبدأ بحرف مختلف . أما إذا كان هناك لوان أو أكثر يبدأ بنفس الحرف (على سبيل المثال إذا كان black أيضاً متاحاً) فإن استخدام الحروف الأولى في بنود القائمة لن يكون ممكناً . ومن الممكن اختيار حروف أخرى بحيث يكون كل بند في القائمة له حرف وحيد . وقد يخلق هذا تشويشاً للمستخدم ولذلك فهو غير مرغوب فيه .

وقد تحتوي بعض البرامج على كل من القائمة الرئيسية وواحدة أو أكثر من القوائم الفرعية . وفي هذه الحالات فقد يكون من المرغوب فيه استخدام كل من الحروف والأرقام كاختيارات من القائمة . وعلى سبيل المثال قد تستخدم الأرقام في القائمة الرئيسية وتستخدم الحروف في القوائم الفرعية . وبين المثال التالي هذه الطريقة .

مثال ١١ - ٤ التمويل الشخصي (حسابات الربح المركب) (Personal Finance (Compound Interest Calculations))

تتطلب كثير من المسائل في اقتصاديات المستهلك بحسابات الربح المركب . وعلى الأخص فقد نرغب أحياناً معرفة المبالغ التي سوف تتراكم لى حساب توفير أو كم يتكلف سداد قرض بعد فترة زمنية معينة بمعدل فائدة محدد ومحدد فترات مضاعفة الربح .

سوف نعتبر في هذا المثال ثلاثة حسابات من هذا النوع وهى :-

١ - القيمة المستقبلية لمبلغ محدد .

٢ - القيمة المستقبلية لإيداعات شهرية ثابتة .

٣ - الأقساط الشهرية لسداد القرض .

سوف نقوم ببناء برنامج لقائمة مفردة يسمح بإنجاز كل هذه الحسابات على حاسب IBM الشخصي وسيكتب البرنامج بلغة ميكروسوفت بيسك .

وفيما يلى المعادلات الضرورية لكل نوع من الحسابات . وهذه المعادلات تستخدم الرموز التالية :

P = القيمة الحالية للمبلغ سواء كان قرض أم ودیعة .

F = جملة المبلغ فى المستقبل .

A = الإيداع الشهرى المنتظم أو السداد الشهرى المنتظم .

i = معدل الفائدة السنوى (معبراً عنه كرقم عشرى) .

m = عدد فترات الفائدة لكل سنة .

n = عدد السنوات .

والمعادلات الحقيقية هى :

١ - القيمة المستقبلية (F) لمبلغ محدد (P)

(أ) مضاعفات :

سنوية ، نصف سنوية ، ربع سنوية ، شهرية ، أو يومية ($m = 1$ أو 2 أو 4 أو 12 أو 365 على الترتيب) .

$$F = P(1 + i/m)^{mn}$$

(ب) مضاعفات مستمرة .

٢ - القيمة المستقبلية (F) لسلسلة الإيداعات الشهرية الثابتة (A)

$$F = Pe^{in}$$

(أ) مضاعفات :-

سنوية ، نصف سنوية ، ربع سنوية ، أو شهرية ($m = 1$ أو 2 أو 4 و 12 على الترتيب) .

$$F = \left(\frac{12A}{m} \right) \left[\frac{(1 + i/m)^{mn} - 1}{i/m} \right]$$

(ب) مضاعفات يومية . ($m = 365$)

$$F = A \left[\frac{(1 + i/m)^{mn} - 1}{(1 + i/m)^{m/12} - 1} \right]$$

(ج) مضاعفات مستمرة

$$F = A \left[\frac{e^{in} - 1}{e^{i/12} - 1} \right]$$

٣ - سداد شهري (A) لقرض (P)

(أ) مضاعفات سنوية او نصف سنوية أو ربع سنوية أو شهرية (m = 1 أو 2 أو 4 أو 12 على الترتيب) .

$$A = \left(\frac{mP}{12} \right) \left[\frac{(1 + i/m)(1 + i/m)^{mn}}{(1 + i/m)^{mn} - 1} \right]$$

(ب) مضاعفات يومية

$$A = P(1 + i/m) \left[\frac{(1 + i/m)^{m/12} - 1}{(1 + i/m)^{mn} - 1} \right]$$

(ج) مضاعفات مستمرة

$$A = Pe^{in} \left[\frac{e^{i/12} - 1}{e^{in} - 1} \right]$$

تصميم البرنامج Design of the Program

سنقوم بتصميم البرنامج التحويلي في طبيعته والعام حيث أمكن في الحدود العملية . وسوف يبدأ البرنامج بعرض قائمة رئيسية تسمح للمستخدم باختيار واحد من الأنواع الثلاثة المختلفة للحسابات أو لإنهاء الحساب . عندئذ سوف تملأ الشاشة والبرنامج سوف يلقن البيانات المطلوب إدخالها . وتستخدم قائمة فرعية لكي تساعد المستخدم في تحديد تكرار إضافة الأرباح .

وبمجرد إدخال البيانات فإن الحسابات المناسبة سوف تجرى ، والإجابة سوف تعرض في أسفل الشاشة . وستظل الشاشة كما هي دون تغيير حتى يقوم المستخدم بالضغط على أى مفتاح يعود عندئذ إلى القائمة الرئيسية .

الخطط التمهيدى للبرنامج The Program Outline

لكي نخطط للبرنامج دعنا نقوم بتعريف المتغيرات الآتية :-

P	=	القيمة الحالية للمبلغ
F	=	القيمة المستقبلية للمبالغ
A	=	الإيداع الشهري المنتظم (أو السداد)
RATE	=	معدل الفائدة السنوى معبراً عنه كنسبة مئوية
I	=	معدل الفائدة السنوى معبراً عنه ككسر (لاحظ أن I=0.01*RATE)
M%	=	عدد فترات الفائدة في العام (بمعنى M%=12 للمضاعفات الشهرية)
N%	=	عدد السنوات

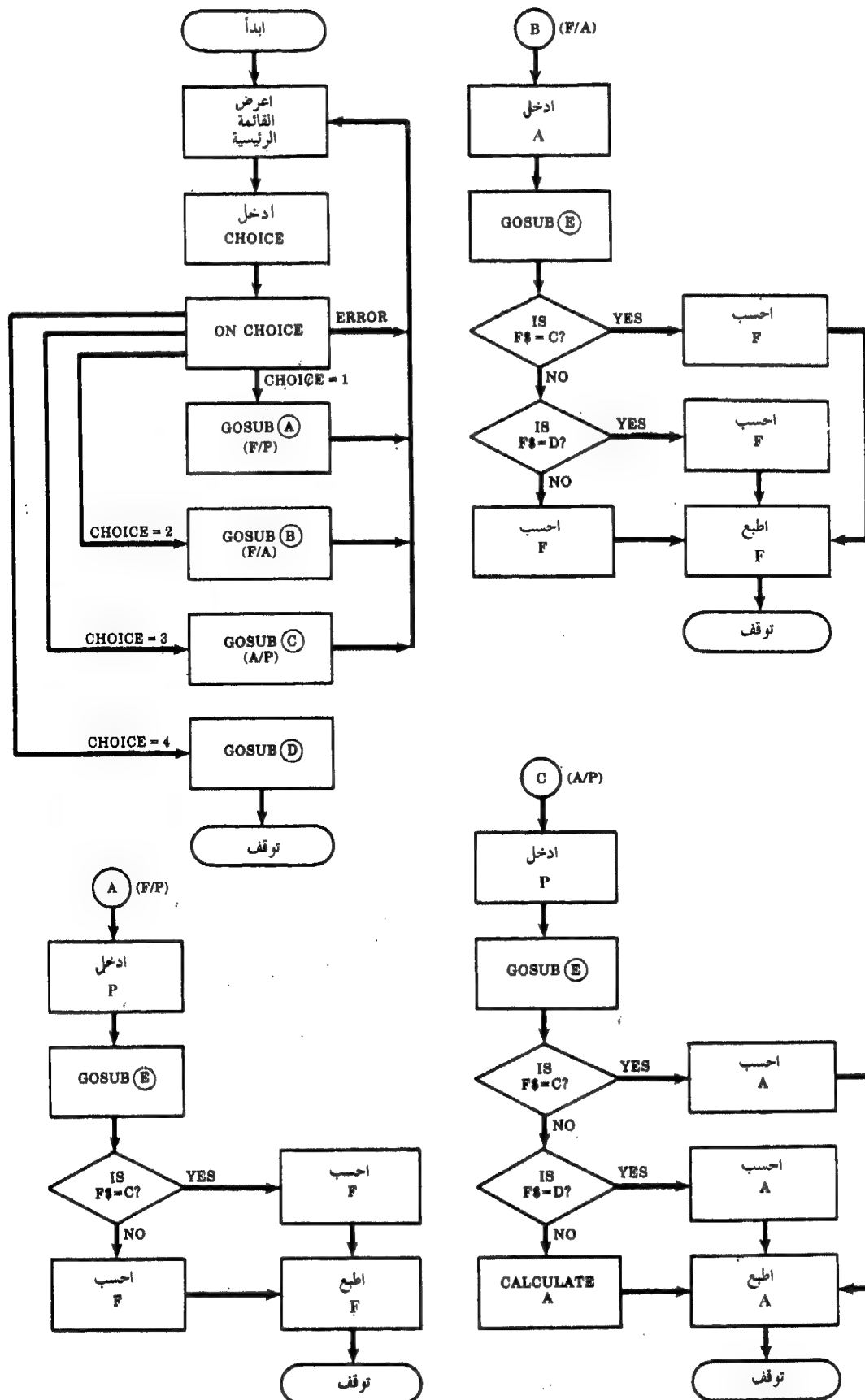
CHOICE = متغير يحدد نوع الحسابات التي تجرى (CHOICE سوف يخصص لها القيمة 1 أو 2 أو 3 أو 4)
 FS = متغير يحدد تكرار المضاعفات (FS سوف يخصص لها "A" أو "S" أو "Q" أو "M" أو "D" أو "C" وذلك للمضاعفات السنوية أو النصف سنوية أو الربع سنوية أو الشهرية أو اليومية أو المستمرة) .

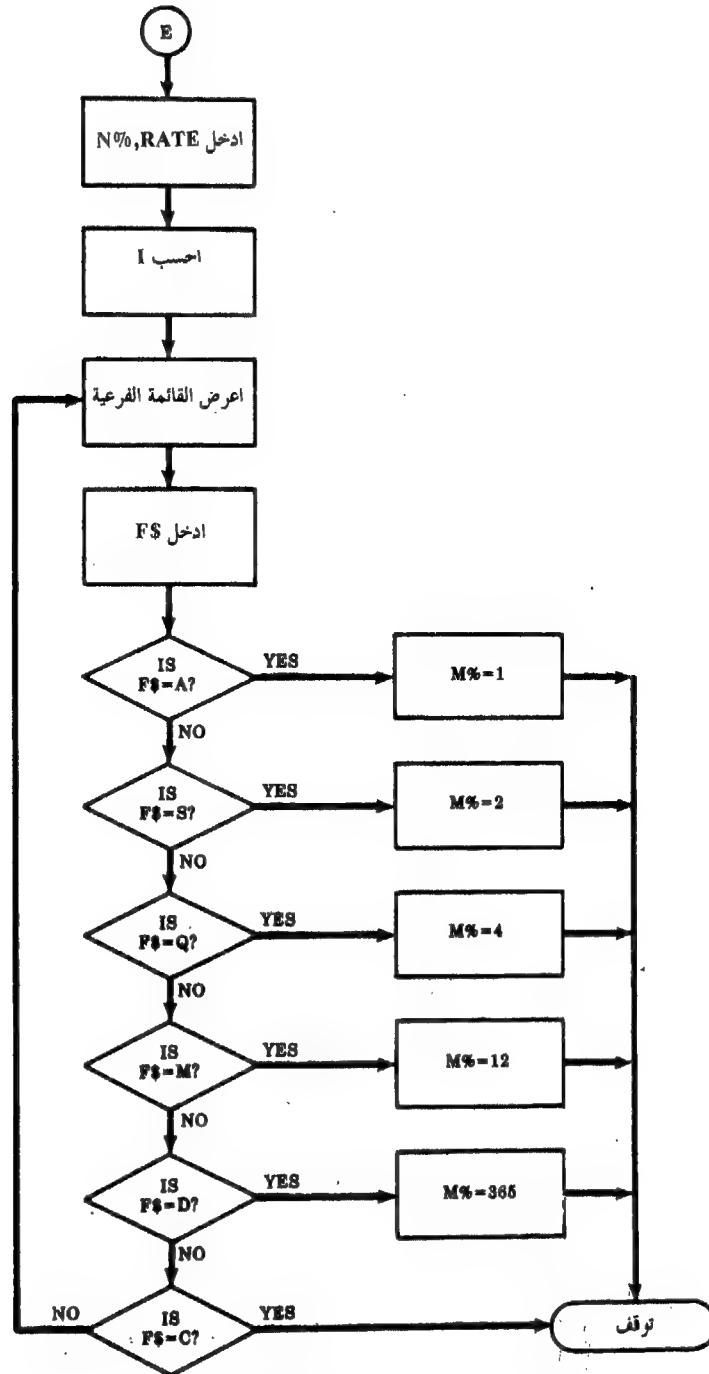
وسوف يعمل البرنامج كما يأتي :-

(١) عرض القائمة الرئيسية

(٢) قراءة قيمة لـ (CHOICE)

(٣) تفرع لبرنامج فرعى مناسب كما تحدده القيمة المخصصة لـ CHOICE





شكل ١١ - ٤ (تكملة)

(أ) CHOICE = 1 (قيمة مستقبلية لمبلغ محدد من المال)

(I) قراءة قيمة P

(II) قراءة قيمة N% و RATE وحساب قيمة I باستخدام المعادلة $I = 0.01 * RATE$

(III) عرض القائمة الفرعية

(IV) قراءة قيمة FS

(V) تخصيص قيمة مناسبة لـ M%

(VI) حساب قيمة F باستخدام المعادلة المناسبة

(VII) عرض قيمة F

(VIII) رجوع للقائمة الرئيسية

(ب) CHOICE = 2 (قيمة مستقبلية لسلسلة من السداد الشهري)

(I) قراءة قيمة A

(II) قراءة قيم N% و RATE وحساب قيمة I باستخدام المعادلة $I = 0.01 * RATE$

(III) عرض القائمة الفرعية

(IV) اقرأ قيمة FS

(V) تخصيص قيمة مناسبة لـ M%

(VI) حساب قيمة F باستخدام المعادلة المناسبة

(VII) عرض قيمة F

(VIII) رجوع للقائمة الرئيسية

(ج) CHOICE = 3 (سداد شهري لقرض)

(I) قراءة قيمة P

(II) قراءة قيمة RATE% وحساب قيمة I باستخدام المعادلة $I = 0.01 * RATE$

(III) عرض القائمة الفرعية

(IV) قراءة قيمة FS

(V) تخصيص قيمة مناسبة لـ M%

(VI) حساب قيمة A باستخدام المعادلة المناسبة

(VII) عرض قيمة A

(VIII) رجوع الى القائمة الرئيسية

(د) CHOICE = 4 (END)

اعرض رسالة إشارة توقف وأنت الحساب .

لاحظ أن الخطوات (II) و (III) و (IV) و (V) هي نفس الأجزاء ٣ (أ) و ٣ (ب) و ٣ (ج) ولذلك فإن التعليمات لهذا الجزء من البرنامج سوف توضع في برنامج فرعي مستقل ومنفصل . وهذا البرنامج الفرعي سوف يحتوى على توليد القائمة الفرعية .
وشكل ١١-٤ يمثل خريطة تدفق تناظر الخطوط العامة السابقة .

برنامج البيسك The BASIC Program

يبين شكل ١١-٥ برنامج البيسك الكامل . لاحظ أن هذا البرنامج طويل نوعاً ما ، ويرجع ذلك لاستخدام الملقنات واختبارات الخطأ والقوائم . وهذه المظاهر السهلة الاستخدام تبسط استخدام البرنامج مع أنه يحتاج لعمل إضافي من جانب المبرمج .
ويبين الشكلان ١١-٦ (أ) و ١١-٦ (ب) جلسة تفاعل نموذجية . ونرى في شكل ١١-٦ (أ) القائمة الرئيسية مع اختيارات المستخدم (تحتها خط) للاختيار رقم 1 (بمعنى قيمة مستقبلية لمبلغ) . الحوار الناتج من هذا الاختيار مبين في شكل ١١-٦ (ب) . لاحظ القائمة الفرعية وعنوانها « تكرار التراكم » والتي تظهر بعد التلقينات للمبلغ الأصلي وعدد السنوات ومعدل الفائدة . ونرى في النهاية أن الإجابة المحسوبة توجد بالقرب من أسفل الشكل . وهكذا نرى أن الاستثمار الأصلي 1000\$ سوف يزداد إلى 3262.04\$ إذا سمح بتراكم الفائدة بمعدل 12 في المئة سنوياً وتتراكم كل ربع سنة لمدة 10 سنوات .

ويوضح الشكلان ٧-١١ و ٨-١١ الحوار الناتج من الاختيارات 2 و 3 على الترتيب . والشكلان ٧-١١ (أ) و ٧-١١ (ب) على الأخص يشيران للقيمة المستقبلية لسلسلة من الإيداعات الشهرية . والشكلان ٨-١١ (أ) و ٨-١١ (ب) يبينان معدل السداد الشهري لقرض .. ويبين شكل ٩-١١ ما يحدث عندما ينتهي البرنامج .

```

10 '***** COMPOUND INTEREST CALCULATIONS *****
20 '
30 KEY OFF: CLS
40 LOCATE 1,1: CLS: PRINT "COMPOUND INTEREST CALCULATIONS"
50 LOCATE 3,1: PRINT " 1 - Future Value of a Given Amount of Money"
60 LOCATE 5,1: PRINT " 2 - Future Value of a Series of Monthly Deposits"
70 LOCATE 7,1: PRINT " 3 - Monthly Loan Repayments"
80 LOCATE 9,1: PRINT " 4 - End"
90 LOCATE 11,1: INPUT "Please enter your selection: ",ANS#
100 CHOICE=VAL(ANS#): IF CHOICE < 1 OR CHOICE > 4 THEN BEEP: LOCATE 11,1:
    PRINT SPACE$(60): GOTO 90
110 ON CHOICE GOSUB 150,270,420,840: GOTO 30
120 '
130 '***** FUTURE VALUE OF A GIVEN AMOUNT OF MONEY (F/P) *****
140 '
150 CLS: LOCATE 1,1: PRINT "FUTURE VALUE OF A GIVEN AMOUNT OF MONEY"
160 LOCATE 3,1: INPUT "Original Amount of Money: $",ANS#
170 P=VAL(ANS#): IF P <= 0 THEN BEEP: LOCATE 3,1: PRINT SPACE$(60): GOTO 160
180 GOSUB 570
190 IF F$="C" OR F$="c" THEN F=P*EXP(I*N%) ELSE F=P*(1+I/M%)^(M%*N%)
200 LOCATE 18,1: PRINT "FINAL AMOUNT =";
210 PRINT USING "***** ,.***";F
220 GOSUB 790
230 RETURN
240 '
250 '***** FUTURE VALUE OF A SERIES OF CONSTANT MONTHLY DEPOSITS (F/A) *****
260 '
270 CLS: LOCATE 1,1: PRINT "FUTURE VALUE OF A SERIES OF MONTHLY DEPOSITS"
280 LOCATE 3,1: INPUT "Amount of Each Payment: $",ANS#
290 A=VAL(ANS#): IF A <= 0 THEN BEEP: LOCATE 3,1: PRINT SPACE$(60): GOTO 280
300 GOSUB 570
310 IF F$="C" OR F$="c" THEN F=A*(EXP(I*N%)-1)/(EXP(I/12)-1): GOTO 350
320 FACTOR=(1+I/M%)^(M%*N%)-1
330 IF F$="D" OR F$="d" THEN F=A*FACTOR/((1+I/M%)^(M%/12)-1): GOTO 350
340 F=(12*A/M%)*FACTOR/(I/M%)
350 LOCATE 18,1: PRINT "FINAL AMOUNT =";
360 PRINT USING "***** ,.***";F
370 GOSUB 790
380 RETURN
390 '
400 '***** MONTHLY LOAN REPAYMENTS (A/P) *****
410 '
420 CLS: LOCATE 1,1: PRINT "MONTHLY LOAN REPAYMENTS"
430 LOCATE 3,1: INPUT "Amount of Money Borrowed: $",ANS#
440 P=VAL(ANS#): IF P <= 0 THEN BEEP: LOCATE 3,1: PRINT SPACE$(60): GOTO 430
450 GOSUB 570
460 IF F$="C" OR F$="c" THEN A=P*(EXP(I*N%)*(EXP(I/12)-1))/(EXP(I*N%)-1):
    GOTO 500
470 FACTOR=(1+I/M%)^(M%*N%)-1
480 IF F$="D" OR F$="d" THEN A=P*(FACTOR+1)*((1+I/M%)^(M%/12)-1)/FACTOR:
    GOTO 500
490 A=(M%*P/12)*(I/M%)*(FACTOR+1)/FACTOR
500 LOCATE 18,1: PRINT "MONTHLY PAYMENT =";
510 PRINT USING "***** ,.***";A
520 GOSUB 790
530 RETURN

```

```

540 '
550 '***** OBTAIN INPUT INFORMATION *****
560 '
570 LOCATE 4,1: INPUT "Number of Years: ",ANS#
580 NX=VAL(ANS#): IF NX <= 0 THEN BEEP: LOCATE 4,1: PRINT SPACE$(60): GOTO 570
590 LOCATE 5,1: INPUT "Annual Interest Rate (Percent): ",ANS#
600 I=.01*VAL(ANS#): IF I <= 0 THEN BEEP: LOCATE 5,1: PRINT SPACE$(60): GOTO 590

610 LOCATE 7,1: PRINT "Frequency of Compounding:"
620 LOCATE 9,1: PRINT "  A - Annual"
630 LOCATE 10,1: PRINT "  S - Semiannual"
640 LOCATE 11,1: PRINT "  Q - Quarterly"
650 LOCATE 12,1: PRINT "  M - Monthly"
660 LOCATE 13,1: PRINT "  D - Daily"
670 LOCATE 14,1: PRINT "  C - Continuous"
680 LOCATE 16,1: INPUT "Please enter your selection: ",F#
690 IF F#="A" OR F#="a" THEN M%=1: RETURN
700 IF F#="S" OR F#="s" THEN M%=2: RETURN
710 IF F#="Q" OR F#="q" THEN M%=4: RETURN
720 IF F#="M" OR F#="m" THEN M%=12: RETURN
730 IF F#="D" OR F#="d" THEN M%=365: RETURN
740 IF F#="C" OR F#="c" THEN RETURN
750 BEEP: LOCATE 16,1: PRINT SPACE$(60): GOTO 680
760 '
770 '***** PAUSE *****
780 '
790 LOCATE 22,1: PRINT "(Press any key to continue)";
800 ANS#=INPUT$(1): RETURN
810 '
820 '***** SIGNOFF *****
830 '
840 LOCATE 13,1: PRINT "GOODBYE, COME AGAIN!"
850 END

```

تابع شکل ١١-٥

COMPOUND INTEREST CALCULATIONS

- 1 - Future Value of a Given Amount of Money
- 2 - Future Value of a Series of Monthly Deposits
- 3 - Monthly Loan Repayments
- 4 - End

Please enter your selection: 1

(a)

FUTURE VALUE OF A GIVEN AMOUNT OF MONEY

Original Amount of Money: \$1000

Number of Years: 10

Annual Interest Rate (Percent): 12

Frequency of Compounding:

- A - Annual
- S - Semiannual
- Q - Quarterly
- M - Monthly
- D - Daily
- C - Continuous

Please enter your selection: Q

FINAL AMOUNT = \$3,262.04

(Press any key to continue)

(b)

شکل ١١-٦

COMPOUND INTEREST CALCULATIONS

- 1 - Future Value of a Given Amount of Money
- 2 - Future Value of a Series of Monthly Deposits
- 3 - Monthly Loan Repayments
- 4 - End

Please enter your selection: 2
(a)

FUTURE VALUE OF A SERIES OF MONTHLY DEPOSITS

Amount of Each Payment: \$100
 Number of Years: 7
 Annual Interest Rate (Percent): 8.5
 Frequency of Compounding:
 A - Annual
 S - Semiannual
 Q - Quarterly
 M - Monthly
 D - Daily
 C - Continuous

Please enter your selection: M
 FINAL AMOUNT = \$11,424.37
 (Press any key to continue)
 (b)

شكل ١١-٧

COMPOUND INTEREST CALCULATIONS

- 1 - Future Value of a Given Amount of Money
- 2 - Future Value of a Series of Monthly Deposits
- 3 - Monthly Loan Repayments
- 4 - End

Please enter your selection: 3
(a)

شكل ١١-٨

MONTHLY LOAN REPAYMENTS

Amount of Money Borrowed: \$5000
 Number of Years: 5
 Annual Interest Rate (Percent): 12.5

Frequency of Compounding:

- A - Annual
- S - Semiannual
- Q - Quarterly
- M - Monthly
- D - Daily
- C - Continuous

Please enter your selection: C

MONTHLY PAYMENT = \$112.66

(Press any key to continue)

(b)

تابع شكل ١١-٨

COMPOUND INTEREST CALCULATIONS

- 1 - Future Value of a Given Amount of Money
- 2 - Future Value of a Series of Monthly Deposits
- 3 - Monthly Loan Repayments
- 4 - End

Please enter your selection: 4

GOODBYE, COME AGAIN!

شكل ١١-٩

١١ - ٣ فحص الخطأ ERROR CHECKING

لقد رأينا في الأقسام السابقة من هذا الفصل العديد من أسئلة فحص الخطأ، وكلها تختص بأخطاء البيانات المدخلة. ومع كل ذلك، فإن هناك أنواعاً أخرى من الأخطاء التي يمكن أن تحدث داخل برنامج البيسك. وهذه تحتوي على أخطاء لغوية (على سبيل المثال FOR بدون NEXT، دليل خارج المدى أو معامل مفقود ... الخ) وأخطاء وقت التشغيل (مثل القسمة على صفر أو قيمة عددية فاقت الحد) وأخطاء الإدخال والإخراج من الأسطوانة المغنطة (مثل ملف غير موجود أو قرص مملوء) وأخطاء الجهاز (مثل الوسيلة غير متاحة أو انتهاء الورق من آلة الطباعة).

وتحتوي معظم نسخ البيسك للحاسب الدقيق على جمل خاصة لتصيد الخطأ والتي يمكنها اكتشاف الأخطاء المعينة عندما تحدث ثم تحول التحكم لنظام تصحيح الأخطاء داخل البرنامج. ويتم هذا عادة بواسطة جملة ON ERROR-GOTO (مثل ON ERROR GOTO 200) كما هو موضح في قسم ٩-٢.

وعادة ما يتم تكويد الأخطاء نفسها في صيغة عددية (مثل الخطأ رقم 11 قد يكون قسمة على صفر). لذا فإنه من الممكن كتابة جملة IF التي تأخذ إجراء معالجاً بمجرد اكتشاف نوع معين من الخطأ (مثل IF ERR = 11 THEN PRINT "Division by zero").

مثال ١١ - ٥

فيما يلي جزء من برنامج ميكروسوفت بيسك الذى يدخل رقمين ويحسب الجذر التربيعى للفرق بينهما ويطبع النتيجة . وبرنامج تصيد الخطأ يولد عبارة ثم يكرر برنامج إدخال البيانات إذا كان الفرق بين الأرقام سالباً .

```
200 ON ERROR GOTO 500
210 INPUT "X="; X
220 INPUT "Y="; Y
230 Z=SQR(X-Y)
240 PRINT "Z="; Z
250 ON ERROR GOTO 0
.
.
.
500 IF ERR=5 AND ERL=230 THEN PRINT "Attempt to take square root
of negative number"; PRINT "Please enter data again"; PRINT;
RESUME 210
```

جملة ON ERROR فى سطر 200 تتسبب فى نقل التحكم إلى سطر 500 إذا تم اكتشاف خطأ (أى خطأ) أثناء تنفيذ البرنامج . جملة IF-THEN فى سطر 500 تختبر الخطأ رقم 5 (استدعاء غير سليم) فى سطر 230 . إذا تم اكتشاف خطأ فإنه يتم توليد رسالة خطأ وينتقل للتحكم عائد إلى السطر 210 وعلى ذلك يسمح للمستخدم بإعادة إدخال البيانات .

(لاحظ أن آخر انتقال للتحكم يتم عن طريق جملة RESUME بدلاً من GOTO . وهناك جملة خاصة RESUME يقصد لاستعمالها عند نهاية برنامج تصيد الخطأ (انظر قسم ٩ - ٢) .

ولاحظ فى النهاية أن جملة ON ERROR—GOTO فى سطر 250 . هذه الجملة تلغى أى اختبار خطأ تال داخل البرنامج . أو بمعنى آخر فإن جملة ON ERROR GOTO 0 تقضى على فاعلية أى تصيد للخطأ والذى قد تم تعريفه سابقاً .

مثال ١١ - ٦ تشغيل درجات امتحانات الطلبة Processing Student Examination Scores

يحتوى شكل ١١ - ١٠ على برنامج ميكروسوفت بيسك لتشغيل درجات امتحانات الطلبة باستخدام ملف بيانات متتابع . هذا البرنامج يختلف عن البرنامج الموجود فى مثال ٩ - ٣٠ (انظر شكل ١٠ - ١٠) .

ويختلف البرنامج الحالى عن البرنامج السابق فى اعتبارات عديدة . على سبيل المثال يسمح البرنامج الحالى للمستخدم أن يحدد أى اسم للملف الذى يريده . (فى النسخة السابقة كان اسم الملف المطلوب SCORES) . وعلى ذلك فإن السطر 80 يلحق المستخدم عن اسم الملف ، والسطر 350 يسبب تخصيص هذا الاسم للملف المطور الجديد الذى تم تحديثه (UPDATE) عند نهاية البرنامج .

ويحتوى البرنامج الحالى بالإضافة إلى ذلك عدة ملقنات واختبارات الخطأ التى لم تكن موجودة فى النسخة السابقة ، فالسطر 160 على سبيل المثال يبين القيمة المسموح بها لرقم الاختبار ، والسطر 250 يبين القيمة المسموح بها لكل درجة امتحان . وجمل الإدخال هذه مصحوبة باختبارات الخطأ (سطر 170 والسطور 260 و 270) بحيث أن القيم المدخلة المستخدمة تقع داخل المدى المبين .

واختبارات الخطأ هى مشابهة لتلك المبينة فى مثال ١١-٢ (شكل ١١-١) . ومع كل فإنه يمكننا الآن استخدام جملة LOCATE CSRLIN-1,1 فى مواضع عديدة . وهذه الجملة تعيد وضع المؤشر (Cursor) لبداية السطر السابق . (لاحظ أن السطور 160 و 250 لا تحتوى جمل LOCATE ، وعلى ذلك فإن إعادة وضع المؤشر لابد أن يكون ضمن برامج فحص الخطأ) .

ويوجد أيضاً اختبار لتحديد إذا ما كان ملف البيانات المعين موجود على القرص المرن الحالى (سطر 390) . وإذا ما كان ملف البيانات غير موجود سوف تعرض رسالة خطأ على الشاشة ويتم تلقين المستخدم ثانياً اسم ملف البيانات . لاحظ أن اختبار الخطأ هذا يتم تنشيطه بواسطة جملة ON ERROR (سطر 70) وهو ما يخالف اختبارات الخطأ الأخرى والتى تختبر البيانات المدخلة مباشرة . لاحظ أيضاً أن اختبار الخطأ هذا ينتهى بجملة RESUME والتى تنقل التحكم ثانياً إلى السطر 80 . ولاحظ فى النهاية أن اختبار الخطأ لاسم الملف ينتهى تنشيطه فى سطر 110 بجملة ON ERROR GOTO 0 .


```

10 '***** PROGRAM TO PROCESS STUDENT EXAMINATION SCORES *****
20 '*****          USING SEQUENTIAL DATA FILES          *****
30 '
40 KEY OFF: CLS
50 DIM C(15)
60 PRINT "UPDATING STUDENT EXAMINATION SCORES"
70 ON ERROR GOTO 390
80 PRINT: INPUT "File name: ",F$
90 OPEN "I",1,F$
100 OPEN "O",2,"UPDATE"
110 ON ERROR GOTO 0
120 INPUT #1,TITLE$
130 INPUT #1,TERM$
140 PRINT: PRINT "Course title: ";TITLE$;SPC(10);"Term: ";TERM$
150 PRINT #2,TITLE$: PRINT #2,TERM$
160 PRINT: INPUT "Exam number (1-15): ",ANS$: K=VAL(ANS$)
170 IF K < 1 OR K > 15 THEN BEEP: LOCATE CSRLIN-1,1: PRINT SPACE$(40):
    LOCATE CSRLIN-2,1: GOTO 160
180 PRINT: INPUT "Calculate averages (Y/N) ";ANS$
190 '
200 '*** BEGIN LOOP ***
210 '
220 PRINT: INPUT #1,N$
230 FOR I=1 TO K-1: INPUT #1,C(I): NEXT I
240 PRINT N$,
250 INPUT "New score (0-100): ",ANS$
260 IF LEFT$(ANS$,1)="0" THEN C(K)=0: GOTO 280 ELSE C(K)=VAL(ANS$)
270 IF C(K) <= 0 OR C(K) > 100 THEN BEEP: LOCATE CSRLIN-1,1: PRINT SPACE$(40):
    LOCATE CSRLIN-1,1: GOTO 240
280 PRINT #2,N$
290 SUM=0
300 FOR I=1 TO K: PRINT #2,C(I): SUM=SUM+C(I): NEXT I
310 IF ANS$="N" OR ANS$="n" THEN 350
320 AVG=SUM/K
330 PRINT "Average =";AVG
340 PRINT #2,AVG
350 IF NOT EOF(1) THEN 220 ELSE CLOSE: KILL F$: NAME "update" AS F$: END
360 '
370 '*** ERROR TRAP FOR INPUT FILE NAME ***
380 '
390 IF ERR=53 AND ERL=90 THEN PRINT: PRINT "File does not exist": RESUME 80
400 END

```

شكل ١١ - ١٠

ويحتوى البرنامج الحالى أيضاً على عدد من ملاحظات إضافية قليلة وعروض على الشاشة بطريقة أفضل تنظيمياً نوعاً ما .

ويبين شكل ١١-١١ جلسة تفاعل نموذجية باستخدام نفس مجموعة البيانات المدخلة (درجات امتحانات الطلبة) كما فى مثال ٩-٣٠ . وهنا أيضاً نجد أن استجابات المستخدم تحتها خط .

وقد يهمننا ان نقارن جلسة التفاعل الحالية بتلك المبينة فى شكل ٩-١١، حيث نلاحظ أن المستخدم قد حدد مبدئياً اسم ملف غير موجود فى الجلسة الحالية (فى السطر الثانى) ، ثم نجد أن البرنامج قد تصيد الخطأ وعندئذ يولد الرسالة «File does not exist» ويلقن ثانياً المستخدم لاسم ملف آخر . والباقي من جلسة التفاعل مشابه للجلسة السابقة فيما عدا التفرقة بين عنوان المقرر والفصل الدراسى والملقبات الممتدة . ومع كل فإن آثار اختبارات الخطأ لرقم الاختبار والدرجات الفردية لا تظهر، ولا بهد للمستخدم أن يقوم بتشغيل البرنامج الحالى حتى يرى تأثير هذه المظاهر .

UPDATING STUDENT EXAMINATION SCORES

File name: score

File does not exist

File name: scores

Course title: Comp Sci 141

Term: Fall 1985

Exam number (1-15): 6Calculate averages (Y/N) ? y

Adams B F New score (0-100): 75
 Average = 71.66666

Brown P New score (0-100): 80
 Average = 65

Davis R A New score (0-100): 55
 Average = 40

Fisher E K New score (0-100): 5
 Average = 4.166667

Hamilton B P New score (0-100): 90
 Average = 91.66666

Jones J J New score (0-100): 80
 Average = 87.5

Ludwig C W New score (0-100): 70
 Average = 53.33333

Osborne T New score (0-100): 80
 Average = 70

Prince W F New score (0-100): 100
 Average = 82.5

Richards E N New score (0-100): 70
 Average = 55

Smith M C New score (0-100): 75
 Average = 67.5

Thomas B A New score (0-100): 10
 Average = 21.66667

Wolfe H New score (0-100): 95
 Average = 64.16666

Zorba D R New score (0-100): 95
 Average = 78.33334

١١-٤ تحقق المستخدم USER VERIFICATION

يقع كل فرد أحياناً في أخطاء عندما يدخل البيانات . ولا تكون عادة نتيجة ذلك خطيرة حيث يمكن أن يقوم المستخدم بتشغيل البرنامج ببساطة باستخدام البيانات الخاطئة ، ثم يقوم بإعادة إدخال البيانات الصحيحة وتشغيل البرنامج مرة أخرى .

ومع كل .. فأحياناً ما تكون النتائج المترتبة على خطأ البيانات المدخلة خطيرة وخصوصاً في حالة البرامج التي تحتوي على طرق إدخال بيانات مطولة أو برامج تحتاج إلى وقت تنفيذ كبير أو برامج تقوم بتخزين البيانات في ملفات البيانات . ويمكن أن تؤدي الحاجة لإعادة إدخال البيانات وإعادة التشغيل يمثل هذه البرامج ضياع الوقت كما تسبب ضيقاً .

ويمكن غالباً التعرف على أخطاء البيانات المدخلة ثم يتم تصحيحها قبل تنفيذ البرنامج ببساطة وذلك بالتحقق من صحة البيانات الحالية . ويتم هذا عادة بواسطة تلقين لبعض مجموعات فرعية لمعالم الإدخال المطلوبة (مثل ما يكفي لشغل كل الشاشة بارتياح) ثم السؤال عما إذا كانت قيم المدخلات صحيحة ، فإذا ما أوضح المستخدم أن البيانات صحيحة ، فإن البرنامج يلقي لأي بيانات إضافية قد تكون مطلوبة أو يستمر في تشغيل البيانات . أما إذا أوضح المستخدم أن البيانات غير صحيحة ، فإنه يطلب من المستخدم أن يعيد إدخال البيانات ويتحقق من صحتها مرة أخرى . وتستمر هذه الطريقة حتى يتم إدخال كل البيانات المطلوبة ويتم التحقق من صحتها .

مثال ١١-٧ تخزين بيانات المعامل Storing Laboratory Data

كتب باحث كيميائي برنامج ميكروسوفت بيسك لتخزين البيانات المأخوذة من عينات عديدة من الفحم في ملف بيانات عشوائى . والمعلومات التالية لكل عينة من الفحم يلزم إدخالها في الحاسب .

- ١ - رقم العينة (عدد صحيح ينحصر بين 1 - 9999) .
 - ٢ - رتبة الفحم (هناك أربعة مدخلات مسموح بها فقط : انتراسيت ، بيتومينوس ، ليجنيت وبيت . وعلى ذلك فإنه يكفي حرف مفرد مثل A ، B ، L أو P لتصنيف رتبة الفحم) .
 - ٣ - النسبة المئوية « رماد » (الشوائب الغير قابلة للاحتراق)
 - ٤ - النسبة المئوية للكبريت .
 - ٥ - النسبة المئوية للرطوبة .
 - ٦ - القيمة الحرارية (سعر حرارى لكل جرام) .
- ولكل عينة فإن طريقة إدخال البيانات تبدأ بالعنوان

COAL SAMPLE DATABASE SYSTEM

وتظهر في أعلى الشاشة الحالية ، ثم تظهر ملفتان فردية واحدة في كل مرة لكل بند من المعلومات المطلوبة . وبعد إدخال كل البيانات يطلب من المستخدم أن يتحقق فيما إذا كانت البيانات صحيحة أو غير صحيحة . (المفروض أن المستخدم سيختار المعلومات التي أدخلها حالياً ولازال معروضة على الشاشة) فإن كان إدخال البيانات صحيحاً فإن المستخدم سيستجيب بالضغط إما على Y (حرف عالى أو منخفض) أو Return . وستسبب هذا في تخزين البيانات المدخلة في ملف عشوائى يسمى SAMPLES مع رقم العينة معبراً عنه كرقم السجل . ثم يتحرك البرنامج لعينة الفحم التالية .

أما إذا أدخلت البيانات غير صحيحة ، فإن المستخدم سيستجيب بالضغط على N (أو أى حرف بخلاف Y أو Return) إذا ما طلب منه التحقق من صحة البيانات ، ثم يتم مسح الشاشة وتظهر الرسالة .

Please reenter the data for this sample

وبعد ذلك مستكرر سلسلة التلقينات فتسمح للمستخدم أن يعيد إدخال (والتحقق مرة أخرى) البيانات لنفس العينة . وتستمر هذه الطريقة كلها حتى يتحقق المستخدم من أن البيانات الحالية صحيحة .

ويوضح شكل ١١ - ١٢ برنامج البيسك الحقيقي ، ويحو سطر 50 أى عرض سابق على الشاشة لتعريفات مفتاح الدالة ، ويفتح السطر 60 ملف البيانات العشوائى (لاحظ أن جملة OPEN مكتوبة بشكل مختلف نوعاً ما عن الموجود فى الأمثلة السابقة التى استخدمت ملفات البيانات العشوائية) . وقد يتم تعريف تركيب كل سجل فى السطر 70 . ثم يتم توليد عنوان الشاشة فى السطر 80 . وتلقن السطور من 100 إلى 120 رقم العينة . وهذا التلقين يوجه المستخدم لإدخال قيمة 0 لكى ينهى البرنامج والقيمة بين 9999 - 1 فيما عدا ذلك . والسطران 110 و 120 يحتويان على برنامج تصيد الخطأ لأى قيم مدخلة التى لا تقع فى هذا المدى .

```

10 '***** COAL SAMPLE DATABASE SYSTEM *****
20 '
30 '**** READ INPUT DATA FOR EACH SAMPLE ****
40 '
50 KEY OFF: CLS
60 OPEN "SAMPLES" AS #1 LEN=20
70 FIELD #1, 1 AS R$, 4 AS A$, 4 AS S$, 4 AS M$, 7 AS C$
80 LOCATE 1,1: PRINT "COAL SAMPLE DATABASE SYSTEM"
90 '
100 LOCATE 3,1: INPUT "Sample number (0 to end, otherwise 1-9999): ",ANS$
110 IF LEFT$(ANS$,1)="0" THEN END ELSE N=VAL(ANS$)
120 IF N <= 0 OR N > 9999 THEN BEEP: LOCATE 3,1: PRINT SPACE$(78): GOTO 100
130 '
140 LOCATE 5,1: INPUT "Rank: A)anthracite, B)bituminous, L)ignite, P)eat: ",ANS$
150 IF ANS$="A" OR ANS$="a" THEN RANK$="A": GOTO 210
160 IF ANS$="B" OR ANS$="b" THEN RANK$="B": GOTO 210
170 IF ANS$="L" OR ANS$="l" THEN RANK$="L": GOTO 210
180 IF ANS$="P" OR ANS$="p" THEN RANK$="P": GOTO 210
190 BEEP: LOCATE 5,1: PRINT SPACE$(78): GOTO 140
200 '
210 LOCATE 7,1: INPUT "Percent ash (0-100): ",ANS$
220 IF LEFT$(ANS$,1)="0" THEN A=0: GOTO 250 ELSE A=VAL(ANS$)
230 IF A <= 0 OR A > 100 THEN BEEP: LOCATE 7,1: PRINT SPACE$(78): GOTO 210
240 '
250 LOCATE 8,1: INPUT "Percent sulfur (0-100): ",ANS$
260 IF LEFT$(ANS$,1)="0" THEN S=0: GOTO 290 ELSE S=VAL(ANS$)
270 IF S <= 0 OR S > 100 THEN BEEP: LOCATE 8,1: PRINT SPACE$(78): GOTO 250
280 '
290 LOCATE 9,1: INPUT "Percent moisture (0-100): ",ANS$
300 IF LEFT$(ANS$,1)="0" THEN M=0: GOTO 330 ELSE M=VAL(ANS$)
310 IF M <= 0 OR M > 100 THEN BEEP: LOCATE 9,1: PRINT SPACE$(78): GOTO 290
320 '
330 LOCATE 10,1: INPUT "Heating value (cal/gm): ",ANS$
340 IF LEFT$(ANS$,1)="0" THEN C=0: GOTO 370 ELSE C=VAL(ANS$)
350 IF C <= 0 THEN BEEP: LOCATE 10,1: PRINT SPACE$(78): GOTO 330
360 '
370 LOCATE 12,1: INPUT "Is everything ok? (Y/N) ",ANS$
380 IF ANS$="Y" OR ANS$="y" OR ANS$="" THEN 440
390 CLS: PRINT "Please reenter the data for this sample"
400 GOTO 100
410 '
420 '**** STORE THE DATA ****
430 '
440 LSET R$=RANK$
450 LSET A$=MKS$(A): LSET S$=MKS$(S): LSET M$=MKS$(M): LSET C$=MKS$(C)
460 PUT #1, N
470 CLS: GOTO 80
480 END

```

وتلقن السطور من 140 إلى 190 رتبة الفحم . وحيث أن هناك أربع قيم فقط مسموح بها للإجابة (A ، B ، L أو P) فإن هذا التلقين يظهر على الشاشة في شكل قائمة مختصرة . ونلاحظ أن المستخدم قد يستجيب باستخدام حروف عالية أو منخفضة .

السطور من 210 إلى 230 تلقن للنسبة المئوية للرماد . ومجموعة الجمل هذه تحتوي على برنامج تصيد الخطأ للقيم العددية التي تقع خارج المدى الموضح من 0 إلى 100 في المائة . ثم تتولد تلقينات مشابهة للنسبة المئوية للكبريت (السطور 250 إلى 270) والنسبة المئوية للرطوبة (السطور 290 إلى 310) والقيمة الحرارية للفحم بالسعر الحرارى لكل جرام (السطور 330 إلى 350) .

وأخيراً فإن تحقق المستخدم يتولد في السطر 370 ، فإذا بين المستخدم أن البيانات صحيحة فإن القيم المدخلة يتم تخزينها في ملف البيانات العشوائى (السطور 440 إلى 460) . ويتم مسح الشاشة ، ويتحول التحكم للسطر 80 مبيئاً بذلك طريقة إدخال البيانات للعينات التالية .

أما إذا أوضح المستخدم أن البيانات غير صحيحة ، فإن رسالة «Please reenter» تظهر السطر 390 . ويتم نقل التحكم إلى السطر 100 ونلاحظ أن البيانات المدخلة لا يتم تخزينها إلا إذا تحققت صحتها .

COAL SAMPLE DATABASE SYSTEM

Sample number (0 to end, otherwise 1-9999): 1330

Rank: A)nthracite, B)ituminous, L)ignite, P)eat: B

Percent ash (0-100): 14.8

Percent sulfur (0-100): 7.0

Percent moisture (0-100): 4.2

Heating value (cal/gm): 6818

Is everything ok? (Y/N) N

(a)

Please reenter the data for this sample

Sample number (0 to end, otherwise 1-9999): 1330

Rank: A)nthracite, B)ituminous, L)ignite, P)eat: B

Percent ash (0-100): 14.8

Percent sulfur (0-100): 7.0

Percent moisture (0-100): 4.2

Heating value (cal/gm): 6188

Is everything ok? (Y/N) Y

(b)

شكل ١١ - ١٣

وبين شكل ١١-١٣ (أ) طريقة إدخال البيانات لعينة فحم نموذجية وكالمعتاد فإن استجابات المستخدم تحتها خط . ونلاحظ أن المستخدم أوضح أن هذه القيم غير صحيحة . وعلى ذلك فإن البرنامج سيتجاهل البيانات الحالية وسيقوم بإعادة طريقة إدخال البيانات لهذه العينة .

ويوضح شكل ١١-١٣ (ب) إدخال البيانات المصححة ونلاحظ أن القيمة الحرارية مختلفة عن ما هو في شكل ١١-١٣ (أ) . وفى هذه المرة أوضح المستخدم أن البيانات صحيحة مما تسبب عنه تخزين بنود البيانات في ملف البيانات العشوائى .

أسئلة للمراجعة Review Questions

- ١١ - ١ لخص الخصائص الأربع للبرمجة السهلة الاستخدام التي تم شرحها في هذا الفصل . ما هو غرض كل منها ؟
- ١١ - ٢ ما هو نوع المعلومات التي تنقل للمستخدم بالتلقين للبيانات المدخلة ؟
- ١١ - ٣ اشرح استخدام طرق فحص الخطأ فيما يتعلق بملقنات البيانات المدخلة . وما هي الوسيلة التي تجعل إحدى هاتين الخاصيتين مكملتين للأخرى ؟
- ١١ - ٤ كيف تختلف القوائم عن الملقنات العادية ؟ ولأى نوع من الحالات تكون القوائم أكثر مناسبة ؟
- ١١ - ٥ ما هي الشروط التي قد تجعل من المرغوب فيه تضمين قائمة رئيسية وواحدة أو أكثر من القوائم الفرعية في نفس البرنامج ؟
- ١١ - ٦ صف فائدة استخدام جملة ON ERROR GOTO وجملة RESUME لإجراء فحص الخطأ . وكيف تختلف طرق فحص الأخطاء هذه عن تلك التي تستخدم فيما يتصل بتلقينات البيانات المدخلة ؟
- ١١ - ٧ هل جملة ON ERROR GOTO و RESUME متاحة على الحاسب الدقيق الخاص بك ؟ إذا لم تكن كذلك فهل هناك جملة مشابهة متاحة ؟
- ١١ - ٨ حدد الطريقة المستخدمة لتكويد الأخطاء على الحاسب الدقيق الخاص بك . هل دوال ERR و ERL متاحة ؟
- ١١ - ٩ صف الطريقة التي يتحقق بها المستخدم من صحة البيانات . ولأى نوع من البرامج يكون تحقق المستخدم مفيداً ؟

مسائل تكميلية Supplementary Problems

المسائل التالية تختص بجمع المعلومات بدلاً من الحل الحقيقي للمسائل . أجب عن الأسئلة باستخدام النسخة الخاصة بك من بيسك الحاسب الدقيق .

- ١١ - ١٠ يصف قسم ١١ - ١ نوعاً من فحص الخطأ الذي فيه يتم اختبار كمية عددية مدخلة لتحديد ما إذا كانت تقع داخل المدى المقبول للقيم أم لا . وهذا النوع من فحص الخطأ يستخدم جملة ودوال ميكروسوفت البيسك الآتية

LOCATE
INPUT (including a prompt)
IF-THEN
BEEP

PRINT
SPACES
GOTO

حدّد ما إذا كانت كل هذه الجمل متاحة على نسختك من بيسك الحاسب الدقيق ، وهل يمكن تفسيرها بنفس الطريقة الموجودة في قسم ١١ - ١ ، وإذا لم يكن كذلك فكيف يمكن إجراء هذا النوع من فحص الخطأ باستخدام نسختك من بيسك الحاسب الدقيق ؟

يبين قسم ١١ - ٣ كيف يتم توليد قائمة باستخدام جمل ودوال ميكروسوفت بيسك التالية :-

١١ - ١١

CLS	IF-THEN
LOCATE	BEEP
PRINT	SPACES
INPUT	GOTO
ANS\$	ON-GOTO
VAL	

حدد ما إذا كانت كل هذه الجمل متاحة على نسختك من بيسك الحاسب الدقيق ؟ وما إذا كان يمكن تفسيرها بنفس الطريقة كما هو مبين في قسم ١١ - ٢ ؟ وإذا لم يكن كذلك فكيف يمكن توليد مثل هذه القوائم باستخدام نسختك من بيسك الحاسب الدقيق .

يصف قسم ١١ - ٣ نوعاً آخر من فحص الخطأ يحتوي على اختبارات خطأ النصوص، وخطأ وقت التشغيل، وخطأ الإدخال/ الإخراج للأقراص وأخطاء الجهاز . وهذا النوع من فحص الأخطاء يستخدم دوال وجمل ميكروسوفت بيسك التالية .

١٢ - ١١

ON ERROR GOTO	IF-THEN
RESUME	PRINT
ERR	ERL

حدد ما إذا كانت كل هذه الجمل متاحة على نسختك من بيسك الحاسب الدقيق وما إذا كان يتم تفسيرها بنفس الطريقة كما هو في قسم ١١ - ٣ ، وإذا لم يكن كذلك فكيف يمكن إجراء هذا النوع من فحص الخطأ باستخدام نسختك من بيسك الحاسب الدقيق ؟ كيف يختلف هذا النوع في فحص الخطأ عن فحص الخطأ الموضح في قسم ١١ - ١ ؟

مسائل للبرمجة

Programming Problems

عدل البرنامج الموجود في مثال ٩ - ١١ (لتوليد أعداد فيبوناتشي والبحث عن الأعداد الأولية) بحيث يحتوي التلقين على عرض المدى المسموح به على الشاشة ، أى ما هو عدد أعداد فيبوناتشي (1-23) . أضف فحص الخطأ الذى يختبر شروط الأخطاء التالية .
(أ) القيم العددية خارج المدى .
(ب) الحروف غير الرقمية .

١٣ - ١١

عدل البرنامج الموجود في مثال ٩ - ٢٧ (البحث عن نهاية عظمى) بحيث تحتوي التلقينات على عرض المدى المسموح به لكل قيمة مدخلة على الشاشة . أضف فحوص الخطأ المناسبة التى تختبر القيم الرقمية خارج المدى والحروف غير الرقمية .
عدل المولد بيجلاتين الموجود في مثال ٩ - ٢٨ بحيث يمكنه إما ترجمة الإنجليزية إلى بيجلاتين أو بيجلاتين إلى الإنجليزية، على أن يحتوي على قائمة بسيطة تسمح للمستخدم أن يختار أيهما .

١٤ - ١١

١٥ - ١١

أعد كتابة البرنامج الموجود في مثال ٤ - ٩ (لحساب الاستهلاك) بحيث يمكن استخدامه مع شاشة عرض TV . ابدأ كل عملية حسابية بقائمة مبنية بها نوع الحسابات المتاحة ، وولد حوار المدخلات والمخرجات المحسوبة أسفل القائمة ، ثم امسح الشاشة فيما بين العمليات الحسابية وتأكد من تضمين فحص الخطأ لكل البيانات المدخلة بما في ذلك اختبارات القائمة .

١٦ - ١١

- ١١ - ١٧ أعد كتابة البرنامج الموجود في مثال ١٠ - ١٢ (متعدد اللغات «hello») بحيث يمكن اختيار لغة معينة عن طريق قائمة تقليدية بدلاً من قلم ضوئي مع تضمين فحص الأخطاء لاختيارات القائمة التي أدخلها المستخدم .
- ١١ - ١٨ حل المسألة ٥ - ٥٦ (مطابقة البلاد مع عواصمها) مستخدماً قائمة لتحديد ما إذا كانت العاصمة موجودة لبلد معين أو البلد موجودة لعاصمة معينة . استخدم مفاتيح الدالة لإجراء اختيارك من القائمة ، وتأكد من تضمين تلقينات وفحص الخطأ مع اختيارات القائمة .
- ١١ - ١٩ اكتب برنامج البيسك الكامل للحاسب الدقيق لخلق واستخدام ملف بيانات عشوائى يحتوى على بيانات لأسماء وعناوين وأرقام وتليفونات كما هو موضح في مسألة ٩ - ٦٥ (انظر أيضاً المسألتين ٨ - ٤٠ و ٩ - ٦٤) وتضمن إمكانية كل الخصائص التالية :-
- (أ) أضف سجلاً جديداً (أى اسم جديد ، عنوان ورقم تليفون) للملف .
- (ب) أوجد وأعرض على الشاشة سجلاً معيناً .
- (ج) احذف سجلاً معيناً .
- (د) اطبع قائمة للملف كله .
- (هـ) ائتم من الحسابات .
- إستخدم فحصاً ثنائياً لايجاد سجلات الأفراد (انظر مثال ٨ - ١٣)
- اجعل البرنامج يولد قائمة تسمح للمستخدم باختيار أحد المظاهر الموضحة عاليه . تأكد من مسح الشاشة ، ثم ارجع للقائمة بعد اجراء كل الملامح المطلوبة مع ضرورة تضمين التلقينات وفحوص الخطأ المناسبة مع كل اختيار للقائمة .

الفصل ١٢

بيانات الحاسب الدقيق Microcomputer Graphics

تسمح معظم الحاسبات الدقيقة بعرض المعلومات بيانياً أو في صورة نصوص . وتسمح مثل هذه العروض بتوليد عدة أنواع مختلفة من الأشكال البيانية والرسومات . وتدعم بعض الحاسبات الدقيقة العروض البيانية المتعددة الألوان . وتسمح بعضها لأنواع معينة من الأشكال البيانية بأن تكون متحركة . والقدره على توليد الألوان والعروض المتحركة والخصنة بتأثيرات الصوت توفر الأساس لأنواع مختلفة من ألعاب الحاسب التي أصبحت شائعة في السنوات الحديثة .

والحقيقة أن كل نسخ يسك الحاسبات الدقيقة تخترى الآن على تعليمات خاصة بالبيانات التي تسمح بتكوين العروض البيانية ببساطة . وعلى سبيل المثال توجد هناك تعليمات فردية لتوليد عدد من الأشكال الشائعة مثل النقاط ، السطور ، المستطيلات ، الدوائر والقطاعات الناقصة . ويمكن استخدام هذه الأشكال لتكوين عروض بيانية مختلفة ومعقدة يمكن أن تتضمن استخدام اللون والصوت . وتتاح أيضا تعليمات خاصة بتوليد عروض متحركة . هذا فضلا عن أنه يمكن التحكم في التحريك باستخدام أجهزة إدخال مساعدة مثل عصا التوجيه والمتجول . ويشرح هذا الفصل استخدام تعليمات يسك الخاصة لعدد يمثل تطبيقات البيانيات .

وكما في الفصول السابقة ، سنركز اهتمامنا على نسخ ميكروسوفت يسك المتقدمة (أى BASICA) والتي يتم تنفيذها على حاسب IBM الشخصي . ومع كل .. فإن كثيرا من الحاسبات الدقيقة الأخرى تستخدم ميكروسوفت يسك ، وعلى ذلك .. فهي تدعم خصائصا مطابقة أو مشابهة لتلك التي سيتم شرحها في هذا الفصل .

١٢-١ أساسيات البيانيات GRAPHICS FUNDAMENTALS

لقد رأينا بالفعل أن النص المعروض يتكون من كلمات تتكون من حروف فردية (أى الحروف والرموز .. الخ) . وعلى ذلك فيمكننا تصور الحروف على أنها العناصر الأساسية لعروض النص . وتوجد حالة مشابهة لعروض البيانيات - حيث تكون العناصر الأساسية هي نقط صغيره تسمى عناصر الصورة (Pixels) . ويمكن تجميع هذه النقط لتكوين أشكال أكثر تعقيدا وذلك كما في حالة تجميع الحروف لتكوين كلمات وجمل وفقرات .

ويقاس مستوى التفصيل (أى التحليل) لعروض البيانيات بدلاله أكبر عدد من النقط الأفقية والرأسية والتي يمكن عرضها في أى وقت واحد . ويمكن تحديد هذه القيم بواسطة أجهزة الحاسب . وتدعم الحاسبات الشخصية العروض البيانية في 640 (أفقى) × 200 (رأسى) من نقط عناصر الصورة (Pixels) ، أو ربما 720 × 350 نقطة (Pixels) وتتغير هذه القيم من حاسب الى آخر . ويمكن الحصول على تحليلات أعلى (أى 1024 × 1024 أو 4096 × 4096) باستخدام أجهزة أعلى ثمنا .

ويحدد ايضا جهاز الحاسب أقصى عدد متاح من الألوان . وعادة لا تدعم الحاسبات الشخصية أكثر من 16 لونا والبعض يدعم عددا أقل . ويمكن الحصول على عدد أكبر من الألوان باستخدام أجهزة أعلى ثمنا . وإذا كانت المستويات المتعددة للتحليل متاحة فإنه كلما ارتفع التحليل قلت الألوان .

وعلى سبيل المثال « يمكن لحاسب IBM الشخصي في شكله الأساسي أن يدعم أسلوبين من البيانيات . ويمكن الإشارة إليهما بالتحليل المتوسط (200x320 مع أربعة ألوان) والتحليل العالي (200x640 بالأسود والأبيض) على الترتيب . ويتم اختيار الأسلوب باستخدام جملة SCREEN . وعلى ذلك فإن بيانيات SCREEN تستدعى أسلوب بيانيات التحليل المتوسط SCREEN 2 أسلوب التحليل العالي . (تذكر أيضا أن SCREEN 0 تستدعى أسلوب النص كما هو مبين في قسم ١٠ - ٦) .

وإذا اخترت بيانات التحليل المتوسط باستخدام جملة SCREEN ، فإنه يمكن اختيار الألوان باستخدام جملة COLOR . ويتم تفسير هذه الجملة بشكل مختلف عن الأسلوب الموجود في النص كما هو مبين في قسم ١٠ - ٦ . ويمكن أن تحتوي جملة COLOR على وجه الخصوص على معلومتين ، تحدد الأولى منها لون الخلفية (تذكر أن المعلومه الأولى تحدد لون الواجهة الأمامية عند وجودها في أسلوب النص) . ويمكن هذا أى قيمة صحيحة بين 0 و 15 معطيا بذلك اختيار 16 لوناً مختلف للخلفية . وتكون الألوان الفردية وما يناظرها من معالم رقمية هي :

0 black	8 gray
1 blue	9 light blue
2 green	10 light green
3 cyan	11 light cyan
4 red	12 light red
5 magenta	13 light magenta
6 brown	14 yellow
7 white	15 high intensity white

وتحدد المعلمة الثانية ، في جملة Color لوحة الألوان Palette فقط . وهى مجموعة من ثلاثة ألوان تعطى بالإضافة إلى لون الخلفية أربعة ألوان يمكن استخدامها في عرض يباين في أى وقت واحد . وهناك فقط لوحتان مختلفتان متاحان . وهما محددتان بالقيم العديدة 0 و 1 على الترتيب . وعلى ذلك .. فإن Color 0,0 سوف تختار خلفيه سوداء (المعلومه الأولى) ولوحة ألوان صفر (المعلمه الثانية) بينما أو Color 0,1 سوف تختار خلفيه سوداء ولوحة 1 . وتكون الألوان الفردية في داخل كل لوحة كالآتي :—

<i>Palette 0</i>	<i>Palette 1</i>
0 background color	0 background color
1 green	1 cyan
2 red	2 magenta
3 brown	3 white

ويتم اختيار لون معين من داخل اللوحة باستخدام إحدى جمل أشكال البيانات أى PSET لنقطة (Pixel) و LINE لسطر أو مستطيل و CIRCLE لدائرة أو قطع ناقص . وسوف نتحدث بإفاضه أكثر حول اختيار لون فيما بعد في هذا الفصل .

مثال ١٢ - ١

نفرض أننا نريد توليد عروض بيانية لتحليل متوسط على حاسب IBM الشخصى بالألوان الأخضر والأحمر والبني المعروضة على خلفية بيضاء . وعلى ذلك فلا بد أن يحتوى برنامج بيسك على التعليمات التالية :—

10 SCREEN 1 : COLOR 7,0

وتحدد التعليمه الأولى (SCREEN 1) بيانات التحليل المتوسط (أى 320 نقطه أفقيه و 200 نقطه رأسيه) . وتحدد التعليمه الثانية COLOR 7,0 لون أبيض للخلفية ثم تختار لوحة 0 لألوان الواجهة الأمامية .

ومن الجدول السابق نرى أن هذه اللوحة تحتوى على الألوان : الأحمر والأخضر والبني .

مثال ١٢ - ٢

نعتبر برنامج بيسك الذى سوف يولد عروضاً بيانية للتحليل العالى على حاسب IBM الشخصى ، فإذا بدأ البرنامج بمسح الشاشة ، فإننا نحتاج للتعليمات الآتية :—

10 KEY OFF : CLS
20 SCREEN 2

وفي هذه الحالة فإن جملة Color غير مطلوبة ، حيث يمكن توليد العروض البيانية عالية التحليل (أى 640 نقطة أفقية و 200 نقطة رأسية) فقط باسود و ابيض (أى أشكال بيضاء مقابل خلفيه سوداء) .

١٢-٢ النقاط والسطور POINTS AND LINES

نسخ ميكروسوفت بيسك التى تستخدم مع حاسب IBM الشخصى تحتوى على جملتى PSET و PRESET اللتان تولدان نقطاً مفردة عند أى موضع معين على الشاشة وبأى لون. ويقصد بأول الجملتين PSET أن تستخدم مع لون يتم اختياره من اللوحة الحالية عندما تكون فى أسلوب بيانات التحليل المتوسط .

ولاستخدام هذه الجملة فلا بد أن يعقب كلمة PSET زوج من المعالم ينحصر داخل قوسين تفصلهما فاصله ، أى (PSET (160, 100) . وتبين هاتان المعلمتان إحداثي النقطة X ، X . ويمكن فى حاله أسلوب التحليل المتوسط أن تنحصر المعلمة الأولى بين 0 و 319 والمعلمة الثانية من 0 إلى 199 . وتمثل النقطة 0 ، 0 أعلى الركن الأيسر للشاشة ، وتمثل النقطة (319,199) أسفل الركن الأيمن .

سيتم زوج الأحداثيات معلمة ثالثة اختيارية تبين لون النقطة أى 2 و (100 و 160) PSET . وقد تنحصر هذه المعلمة بين 0 و 3 . وفى بيانات التحليل المتوسط يتم تفسير قيمة هذه المعلمة كأحد الألوان من اللوحة الحالية النشيطة (أنظر قسم ١٢ - ١) . وإذا لم تتضمن جملة PSET صراحة هذه المعلمة الأخيرة فإن اللون رقم 3 سوف يتم اختياره اتوماتيكياً .

مثال ١٢ - ٣

يحتوى برنامج ميكروسوفت بيسك مكتوب على حاسب IBM الشخصى على الجمل التالية :

10 SCREEN 1 : COLOR 0,1

20 PSET (160,100),2

ويحدد السطر 10 بيانات تحليل متوسط بخلفيه سوداء و لوحة ألوان رقم 1 (الألوان : سماءى و بنفسجى وأبيض) والسطر 20 يتسبب فى توليد نقطة بنفسجية فى مركز الشاشة .

مثال ١٢ - ٤

يحتوى شكل ١٢ - ١ على برنامج ميكروسوفت بيسك مكتوب لحاسب IBM الشخصى والذى يولد 100 نقطة مختلفة فى أوضاع عشوائية على الشاشة .

وتستخدم بيانات تحليل متوسط وتعرض كل نقطة بلون مختار عشوائياً من اللوحة رقم 1 (باستثناء عدم اختيار لون الخلفية حيث أن مثل هذه النقطة تكون غير مرئية) .

```
10 KEY OFF : CLS
20 SCREEN 1 : COLOR 0,1
30 FOR I=1 TO 100
40   X=INT(320*RND)
50   Y=INT(200*RND)
60   CLR=1+INT(3*RND)
70   PSET(X,Y),CLR
80 NEXT I
90 END
```

شكل ١٢ - ١

وينقل سطر 10 أية عروض سابقة لتعاريف مفتاح الدالة ثم يسمح الشاشة . ويحدد سطر 20 بيانات تحليل متوسط بخلفية سوداء و لوحة ألوان رقم 1 .

ويتم توليد النقاط الفردية باستخدام الحلقة التكرارية FOR - TO التى تحتوى على السطور 30 الى 80 . ويولد السطران 40 و 50 زوجاً من إحداثيات عشوائية . ويولد سطر 40 رقماً عشوائياً تقع قيمته بين 0 و 319 ، ويولد سطر 50 رقماً عشوائياً بين 0 و 199 . ويتم اختيار اللون

عشوائيا في سطر 60 حيث يتم توليد رقم عشوائى تقع قيمته بين 1 و 3 . وفى النهاية فإنه فى الحقيقة يتم توليد كل نقطة فى الحقيقة فى سطر 70 باستخدام القيم الحالية التى تم توليدها عشوائيا .

ونشجع القارئ على تنفيذ هذا البرنامج وملاحظة ما يحدث (ولابد من مشاهدة العروض التى يتم توليدها لبرامج البيانات حتى يمكن تفهمها) .

اعتبر استخدام جملة PSET فى أسلوب بيانات التحليل العالى . وقد يتراوح إحداثا x و y من 0 الى 639 ومن 0 الى 199 على الترتيب . وعلى ذلك فالنقطة (0 ، 0) تمثل أعلى الركن الأيسر من الشاشة . والنقطة (639 , 199) تمثل أسفل الركن الأيمن .

وسوف تفسر معلمة اللون بشكل مختلف عنه فى أسلوب التحليل المتوسط . فالقيمة الزوجية (أى 0 أو 2) ستبين الأسود ، والقيمة الفردية (أى 1 أو 3) ستبين الأبيض . وإذا كان هذا لا يتضمن معلمة اللون ، فإن اللون رقم 1 (أبيض) سيتم اختياره .

مثال ١٢ - ٥

يحتوى برنامج ميكروسوفت بيسك مكتوب لحاسب IBM الشخصى على الجمل الآتية :-

```
10 SCREEN 2
20 PSET (320,100)
```

يحدد سطر 10 بيانات التحليل العالى (ومع لون أبيض على خلفية سوداء) والسطر 20 يسبب توليد نقطة بيضاء عند مركز الشاشة . ومن الأمور الشيقة أن نقارن هذه الجمل مع مجموعة الجمل المشابهة فى مثال ١٢ - ٣ لاحظ أن المثال الحالى لا يحتوى على جملة COLOR . لاحظ أيضا أن معلمة اللون لا تتضمنها جملة PSET .

مثال ١٢ - ٦

يبين شكل ١٢ - ٢ تغييراً فى البرنامج المعروض فى مثال ١٢ - ٤ ومع كل .. فإنه يتم توليد 100 نقطة عشوائية فى بيانات التحليل العالى . والمنطق أساسا هو نفس ما سبق عرضه ، مع تغيرات صغيرة فى بعض الجمل حتى يلائم أسلوب بيانات التحليل العالى (لاحظ حذف جملة COLOR فى البرنامج الحالى) .

```
10 KEY OFF : CLS
20 SCREEN 2
30 FOR I=1 TO 100
40   X=INT(640*RND)
50   Y=INT(200*RND)
70   PSET(X,Y)
80 NEXT I
90 END
```

شكل ١٢ - ٢

ونشجع القارئ على تنفيذ هذا البرنامج ومقارنة العرض البياى مع ما تم توليده فى مثال ١٢ - ٤ .

دعنا الآن نغير اهتمامنا الى PRESET وهى الجملة الثانية من ميكرو سوفت بيسك التى تولد نقطة مفردة . وهذه الجملة تطابق PSET فيما عدا تفسير اللون البديل الافتراضى (أى اللون الذى يتم اختياره تلقائيا إذا كانت معلمة اللون غير موضحة صراحة) . وبينما تقوم جملة PSET باختيار اللون رقم 3 تلقائيا بواسطة البديل الافتراضى ، فإن جملة PRESET سوف تختار تلقائيا لون الخلفية وعلى ذلك فإنه قد يكون مناسباً فى بعض التطبيقات استخدام PSET لتوليد النقط و PRESET لمسح النقط (وذلك بإعادة توليد النقط فى لون الخلفيه) .

مثال ١٢ - ٧

اعتبر الجمل التالية التى يتضمنها برنامج ميكروسوفت بيسك مكتوب لحاسب IBM الشخصى .

```

10 SCREEN 1 : COLOR 2,0
:
:
100 PSET (160,100),2
110 FOR I=1 TO 2000 : NEXT I
120 PRESET (160,100)

```

يحدد سطر 10 بيانات تحليل متوسط بخلفية خضراء ولوحة ألوان رقم 0 (ألوان أخضر، أحمر وبني). ويتسبب سطر 100 في توليد نقطة حمراء عند مركز الشاشة. ويتم توليد تأخير وقت مختصر بواسطة الحلقة التكرارية الفارغة FOR — TO في السطر 110. ويتبع هذا التأخير الوقتي مسح النقطة الحمراء (أى إعادة توليدها في لون الخلفية) في سطر 120

مثال ١٢ — ٨ نقط في الفراغ Dots in Space

نرى في شكل ١٢ — ٣ برنامج ميكرو سوفت بيسك أكثر شمولاً لحاسب IBM الشخصى الذى يُبنى على بعض الأفكار المعروضة في الأمثلة السابقة. ويولد هذا البرنامج 200 نقطة مختلفة في أماكن عشوائية على الشاشة باستخدام بيانات التحليل المتوسط. ويتم اختيار كل نقطة عشوائياً من اللوحة رقم 1.

ومع كل تظل النقاط مرئية لفترة محدودة من الوقت فقط. وأخيراً فإن كل نقطة تختفى (يتم مسحها) وتستبدل بنقطة جديدة في موضع آخر على الشاشة. والتأثير الناتج هو شاشة ممتلئة دائماً بنقط ملونة ومتباعدة عشوائياً، إلا أنها تتغير باستمرار كنقط قديمة تختفى وأخرى جديدة تظهر.

```

10 REM *** DOTS ***
20 '
30 SCREEN 1 : COLOR 0,1
40 DIM X(200),Y(200)
50 RANDOMIZE
60 KEY OFF : CLS
70 I=1
80 '
90 '*** begin main loop ***
100 '
110 PRESET(X(I),Y(I))
120 X(I)=INT(320*RND) : Y(I)=INT(200*RND)
130 CLR=1+INT(3*RND)
140 PBET(X(I),Y(I)),CLR
150 I=I+1
160 IF I > 200 THEN I=1
170 GOTO 110
180 END

```

شكل ١٢ — ٣

ودعنا الآن نعتبر أساس عمل هذا البرنامج سطرًا بسطر. يحدد سطر 30 بيانات تحليل متوسط بخلفية سوداء ولوحة ألوان رقم 1 (سماوى، بنفسجى وأبيض). وجملته DIM في سطر 40 تؤدي إلى تخزين 200 زوج من الإحداثيات (إحداثيات كل النقط يتم عرضها في أى وقت واحد). وسطر 50 يستعمل مولد الأعداد العشوائية، وسطر 60 ينقل أى عرض سابق لتعريفات مفتاح الدالة ويمسح الشاشة، ثم تستعمل الحلقة التكرارية ذلك في سطر 70.

وتحتوى الحلقة التكرارية الرئيسية على السطور 110 إلى 170. ويتسبب سطر 110 في مسح النقطة الحالية (أى النقطة رقم 1). ونحصل بعد ذلك على مجموعة جديدة من الإحداثيات للنقطة رقم 1 في سطر 130. ويتسبب سطر 140 في توليد النقطة الجديدة رقم 1. لاحظ أن الإحداثيات الجديدة سوف يتم تخزينها في المجموعتين المتراصتين X و Y. وهذا يسمح باستدعاء النقطة الحالية ومسحها في وقت لاحق. ومع كل لا يحدث هذا المسح حتى يتم توليد 199 نقطة إضافية.

وفي النهاية ، يزداد العدد في سطر 150 ويعاد وضع قيمته إلى 1 إذا كان ذلك ضروريا في سطر 160 وتسمح هذه الطريقة باستمرار الحلقة التكرارية بغير حدود .

ونهيى بالقارئ أنيا بتنفيذ هذا البرنامج وملاحظة ما يحدث لكي يكتسب تفهما عاليا للتأثير الديناميكي الذي ينشأ .
وتحتوى نسخة ميكروسوفت بيسك المتاحة لحاسب IBM الشخصى على جملة LINE التي تسمح برسم خط مستقيم بين نقطتين على الشاشة . ويمكن تحديد كل من النقطتين صراحة ، أو يمكن أن تكون إحداها هي النقطة الأخيرة المشار إليها بواسطة جملة بيانات سابقة .
ويجب أن يتبع كلمة LINE في أول صورة لها زوجان من المعالم حيث يمثل كل زوج الإحداثيين Y,X لإحدى النقطتين ويمنحصر الإحداثيان بين قوسين وتفصلهما فاصلة . ويجب فصل النقطتين بواسطة شرطة (علامة —) . أى . (300 , 150) - (20 , 50) . وتعتمد قيم الإحداثيات المسموح بها على أسلوب معين للبيانات (تحليل متوسط أو عالى) كما تم شرحه في هذا القسم .
ويمكن أن يتبع الإحداثيات معلمة اختيارية تبين لون السطر مثل 1 و (300,150) - (20,50) LINE . والقيمة المخصصة لهذه المعلمة يمكن أن تنحصر بين 3,0 . وتفسر بنفس الطريقة كقيمة معلمة اللون في جملة PSET .

مثال ١٢ — ٩

يحتوى برنامج ميكروسوفت بيسك مكتوب لحاسب IBM الشخصى على الجمل التالية :

```
10 SCREEN 1 : COLOR 4,1
20 LINE (20,50)-(300,150),3
```

يحدد سطر 10 بيانات تحليل متوسط بخلفية حمراء ولوحة ألوان رقم 1 (ألوان : سماوى ، وبنفسجى وأبيض) . ويولد سطر 20 قطر أبيض يبدأ في أعلى اليسار أى النقطة (20,50) إلى أسفل اليمين ، أى النقطة (300,150) .

لاحظ أن المعلمة الأخيرة كان في الامكان حذفها في جملة LINE ، أى :

```
20 LINE (20,50)-(300,150)
```

حيث أن اللون رقم 3 يتم اختياره تلقائيا إذا لم تبين القيمة صراحة . وتسمح الصورة الثانية لجملة LINE بحذف الزوج الأول من الإحداثيات ، أى LINE — (300,150) وتسبب هذه الجملة في رسم خط النقطة التي أشير إليها أخيرا (في جملة سابقة) للنقطة المعنية حاليا . وتكون هذه الصورة من جملة LINE مفيدة عند رسم سلسلة من الخطوط المتصلة .

مثال ١٢ — ١٠ السهم المفاجيء A Lightning Bolt

يبين شكل ١٢ — ٤ برنامج ميكروسوفت بيسك قصيرا مكتوبا لحاسب IBM الشخصى الذى يتسبب في عرض « سهم مفاجيء » أحمر على الشاشة .

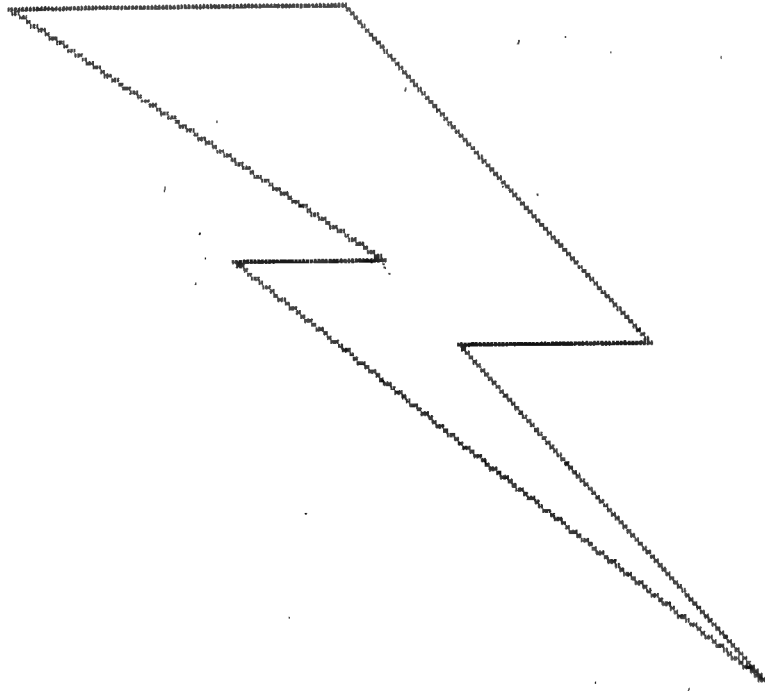
```
10 ***** LIGHTNING BOLT *****
20
30 KEY OFF : CLS
40 SCREEN 1 : COLOR 0,0
50 LINE (20,20)-(120,80),2 : LINE -(80,80),2
60 LINE -(220,180),2 : LINE -(140,100),2
70 LINE -(190,100),2 : LINE -(110,20),2
80 LINE -(20,20),2
90 END
```

ومسح سطر 30 الشاشة ويختار سطر 40 بيانات التحليل المتوسط بخلفية سوداء وبلوحة ألوان رقم 0 (أخضر وأحمر وبني) . وتحتوى السطور 50 إلى 80 على سبع جمل LINE مولدة بذلك سبعة خطوط متصلة تكون السهم المفاجيء . لاحظ أن كلا من صورتى جملة LINE قد تم استخدامها . ويجب استخدام الصورة الأولى لتوليد الخط الأول . ومع كل .. فبمجرد إيجاد النقطة الأخيرة تستخدم الصورة الثانية من جملة LINE لتوليد الخطوط الباقية .

وينفذ البرنامج يتم توليد الصورة المبينه فى شكل ١٢ — ٥ على الشاشة بخطوط حمراء على خلفيه سوداء . وتدعم بعض الحاسبات الدقيقه صوراً أخرى من جملة LINE التى تسمح لمثل هذه الخصائص مثل تحديد خط بالنسبه الى نقطه سابقه بدلاً من نقطه مطلقه الى أخرى ، أو توليد خط متقطع . ولن نناقش هذه الخصائص حيث انها تستخدم فى حالات أقل من الخصائص التى سبق شرحها .

وعلى القراء المهتمين الرجوع الى كتيبات برجه اليسك التى تصاحب حاسباتهم الخاصه لمعلومات اضافيه لهذه الخصائص .

ويمكن استخدام جملة LINE المفرده ايضا لتوليد مستطيل كامل بدلاً من خط مفرد وسنرى كيف يتم ذلك فى قسم ١٢ — ٣ .



شكل ١٢ — ٥

مثال ١٢ — ١١ الخطوط المتحركة (الفن الحركى) Moving Lines (Kinetic Art)

دعنا الآن نعيد انتباهنا لبرنامج ميكروسوفت بيسك يعتمد على نفس الفكره العامة للبرنامج المبين فى مثال ١٢ — ٨ ، ولكنه يولد عروضاً أكثر اثارة .

ونريد توليد سلسلة من الخطوط الملونة فى بيانات التحليل المتوسط ، موزعة عشوائيا حول الشاشة . ومع كل .. فإن الخطوط المتتابعة لن يتم توليدها مستقلة واحدة عن الأخرى ، بل إن نقطتها النهائية المتوالية ستختلف بإزاحات صغيرة . وسيتم عرض مجموعات من الخطوط المتتالية بلون مفرد يتم اختياره عشوائيا من لوحة اللون رقم 1 . وسيتم تغيير كل من هيمه الإزاحات واللون دوريا كلما استمر البرنامج فى التنفيذ . وسيقوم البرنامج دائما بعرض نفس عدد الخطوط على الشاشة ولكن سيتم مسح كل خط وتوليد خط جديد .

وعلى ذلك فستظهر الخطوط لتحرك حول الشاشة عشوائياً ولكن فى نموذج مرتبط بها وبألوان مختلفة . وبالإضافة إلى ذلك سننشأ أحيانا نماذج إضافية متموجه مشوقه كلما تقاطعت الخطوط الجديده مع الخطوط الموجوده . والتاثير النهاى هو أحد المشاهد الجميلة المدهشه .

ويحتوى شكل ١٢ - ٦ على البرنامج الحقيقى . وسطر 30 يستدعى بيانات التحليل المتوسط ثم يختار خلفيه سوداء ولوحه ألوان رقم 1 (الألوان سماوى ، بنفسجى وابيض) . ويعرف سطر 40 كل متغيرات البرنامج كمتغيرات من نوع الأعداد الصحيحة . ويقوم سطر 50 بتخزين 150 سطر أى زوج من النقط النهائية . ويبدأ مولد الأرقام العشوائيه فى سطر 60 . ويتم مسح الشاشة فى سطر 70 . ويعطى سطر 80 زوجاً ابتدائياً من النقط النهائية (أى (X1,Y1) ، (X2, Y2) على الترتيب) للخط الأول .

ويعطى سطر 90 قيمه ابتدائية للعدادات POINTCOUNT, COLORCOUNT . والأخيرة من هذه ، I ، هى ببساطة عداد حلقة تكرارية يزداد أثناء كل مرور خلال الحلقة التكرارية . وهذا العدد يتراوح بين 1 و 150 ، ويعاد وضعه إلى 1 متى زادت قيمته عن 150 . ويحتاج العدادان الأخيران لبعض الشرح الإضافى .

فمثلا COLORCOUNT هو عداد يحدد متى (ما الذى يمر خلال الحلقة التكرارية) يتم اختيار لون مختلف من اللوحة . ويتم وضع COLORCOUNT بداية عند الصفر فى سطر 90 ، بحيث يمكن توليد قيمة غير صفريه جديدة عشوائيا أثناء المرور الأول خلال الحلقة التكرارية (فى سطر 190) . وهذا يصاحب قيمة جديدة مولدة عشوائيا للون (أيضا فى سطر 190) . وأثناء كل مرور خلال الحلقة التكرارية تتناقص COLORCOUNT بمقدار 1 . وعندما تصبح COLORCOUNT مرة أخرى مساوية للصفر تتولد قيمة جديدة ، ويتم اختيار لون جديد ، وهكذا .

```

10 REM ***** LINES *****
20 '
30 SCREEN 1: COLOR 0,1
40 DEFINT A-Z
50 DIM X1(150),Y1(150),X2(150),Y2(150)
60 RANDOMIZE
70 KEY OFF: CLS
80 X1=120: Y1=70: X2=200: Y2=130
90 COLORCOUNT=0: POINTCOUNT=0: I=1
100 '
110 '***** BEGIN MAIN LOOP *****
120 '
130 '*** erase old line ***
140 '
150 LINE (X1(I),Y1(I))-(X2(I),Y2(I)),0
160 '
170 '*** generate new values for counters if required ***
180 '
190 IF COLORCOUNT=0 THEN COLORCOUNT=5*(1+INT(10*RND)): CLR=1+INT(3*RND)
200 IF POINTCOUNT=0 THEN POINTCOUNT=5*(1+INT(10*RND)): DX1=INT(9*RND)-4:
    DY1=INT(9*RND)-4: DX2=INT(9*RND)-4: DY2=INT(9*RND)-4
210 '
220 '*** generate end points for new line ***
230 '
240 X1=X1+DX1: IF X1 < 0 OR X1 > 319 THEN X1=X1-2*DX1
250 Y1=Y1+DY1: IF Y1 < 0 OR Y1 > 199 THEN Y1=Y1-2*DY1
260 X2=X2+DX2: IF X2 < 0 OR X2 > 319 THEN X2=X2-2*DX2
270 Y2=Y2+DY2: IF Y2 < 0 OR Y2 > 199 THEN Y2=Y2-2*DY2
280 '
290 '*** display new line, then save end points ***
300 '
310 LINE (X1,Y1)-(X2,Y2),CLR
320 X1(I)=X1: Y1(I)=Y1: X2(I)=X2: Y2(I)=Y2
330 '
340 '*** adjust counters and repeat ***
350 '
360 I=I+1
370 IF I > 150 THEN I=1
380 COLORCOUNT=COLORCOUNT-1: POINTCOUNT=POINTCOUNT-1
390 GOTO 150
400 END

```

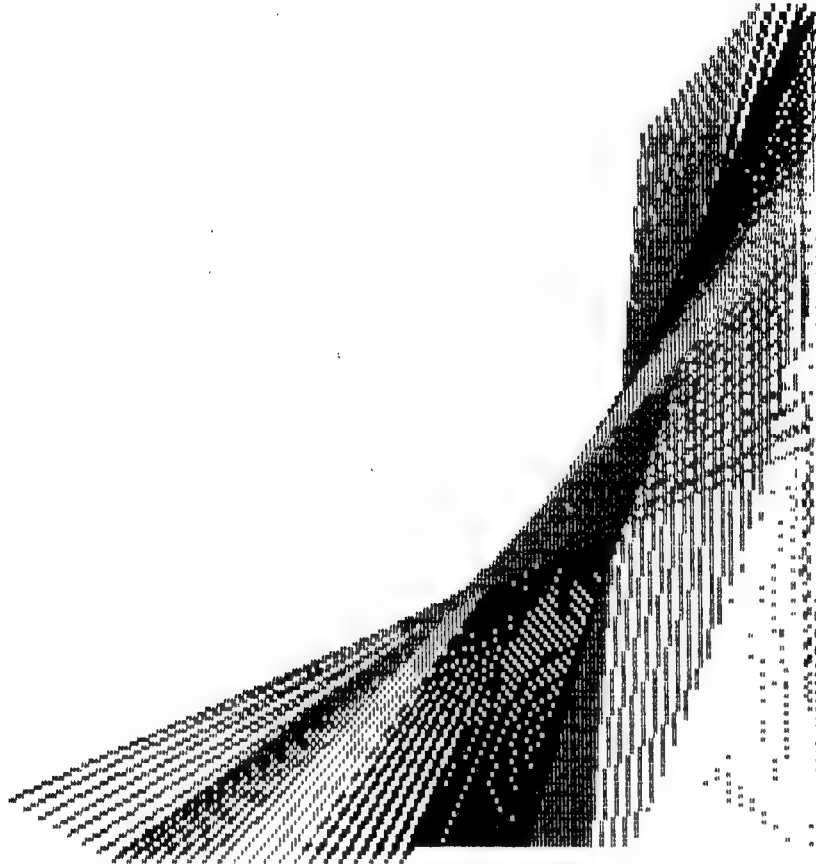

وبالمثل فإن POINTCOUNT هو عداد يحدد متى يتم اختيار قيم مختلفة لتغير إزاحة الخطوط المتتابة . ويتم وضع POINTCOUNT بداية عند الصفر في سطر 90 بحيث يتم توليد قيمه غير صفريه أثناء المرور الأول خلال الحلقة التكرارية (في سطر 200) . هذا بالإضافة إلى أنه سيتم تحديد مجموعة جديدة من القيم لكل من DX1, DY1, DX2, DY2 في سطر 200 وتحدد هذه المعالم الأربعة الإزاحة للأزواج المتتابة في النقط النهائية والتي تحدد بدورها مواقع الخطوط المتتالية . يتم تناقص POINTCOUNT أثناء كل مرور خلال الحلقة التكرارية بمقدار 1 . وعندما تصبح POINTCOUNT مرة أخرى مساوية للصفر ، يتم توليد قيمة جديدة ويتم تحديد مجموعة جديدة من الإزاحات .

ويتراوح الجزء المعاد من البرنامج في السطور 150 إلى 390 . ويتسبب سطر 150 في فسخ السطر المعرف بمجموعة النقط النهائية التي ترتيبها I وتخزن حاليا في عناصر الصف X1 (I) ، Y1 (I) ، X2 (I) ، Y2 (I) . ومع كل ، أثناء 150 المرور الأولى خلال الحلقة التكرارية لن يكون هناك مثل هذه النقط النهائية . ويختار سطر 190 ما إذا كان COLORCOUNT يساوى صفرا . وإذا كان كذلك فإنه يتم اختيار قيمة جديدة عشوائية تتراوح من 5 إلى 50 . وقيمة جديدة للون تتراوح من 1 إلى 3 يتم توليدها عشوائيا .

وسيطل اللون كما هو دون تغير حتى تصبح COLORCOUNT مساوية للصفر مرة أخرى .

ويحتوى سطر 200 على اختيار مماثل POINTCOUNT ، فإذا كانت تساوى صفرا فإنه يتم توليد قيمة جديدة تتراوح من 5 إلى 50 عشوائيا . ويتم اختيار مجموعة جديدة من القيم لمعالم الإزاحة . ويتم تخصيص قيمه منفصلة لكل من هذه المعالم ، تتراوح من -4 إلى +4 . وستظل معالم الإزاحة كما هي دون تغيير حتى تصبح POINTCOUNT مساوية للصفر مرة أخرى .

وتحسب السطور من 240 إلى 270 زوج جديد من النقط النهائية للخط الحالى بالنسبة إلى النقط النهائية للخط السابق . وسيتم التحقق لكل من الإحداثيات الجديدة لتحديد ما إذا كانت قيمتها الجديدة أكبر أو أصغر من اللازم (مسبة بذلك وضع النقط النهائية بعد حواف الشاشة) . وإذا تحقق هذا فيتم ضبط الإحداثى تبعا لذلك .



ويسبب سطر 310 عرض الخط الجديد على الشاشة ، ويسبب سطر 320 تخزين النقط النهائية لهذا الخط في صفوف متراسه وهذا يسمح باستدعاء الخط الحالي ومسحه في زمن لاحق ، ومع كل ، لن يتم هذا المسح حتى يتم توليد 149 سطر إضافي .
وتكمل السطور 390 الحلقة التكرارية وذلك بضبط قيم المعالم ثم العودة الى سطر 150 لبدء مرور آخر . لاحظ أن الحلقة التكرارية سوف تستمر في التنفيذ بغير حدود حيث أنه لا يوجد شرط معين للإيقاف .
وعند تنفيذ البرنامج يتم توليد نماذج مشابهة لتلك المبينة في شكل ١٢ - ٧ في عرض يتحرك باستمرار حول الشاشة . وعلى القارئ أن يلاحظ حقيقة البرنامج أثناء تنفيذه على شاشة ملونة حتى يمكنه أن يتفهم كاملا التأثيرات المدهشة التي تتكون .
وقد تم توجيه الأمثلة المعروضة في هذا الفصل حتى الآن نحو استخدام البيانات لإحداث تأثيرات مسلية أو فنية . ومع كل .. فإنه يمكن أيضا استخدام البيانات بفاعلية كبيرة لتوليد عروض مفيدة في التجارة والتطبيقات الفنية . ويتضح ذلك في المثال التالي .

مثال ١٢ - ١٢ الانحدار الخطي مع العرض البياني Linear Regression with Graphical Display

دعنا الآن ندرس مسألة توفيق خط مستقيم لمجموعة نقط بيانات معطاه باستخدام طريقة المربعات الصغرى كما هو مبين في مثال ٧ - ٢٢ .
(هذه المسألة يشار إليها عادة بالانحدار الخطي *) . وسوف نعرض برنامج ميكروسوفت بيسك كتب لحاسب IBM الشخصي الذي يقوم أولا بالحسابات الرياضية الضرورية ثم يولد عرض بياني لنقط البيانات الفردية وخط الانحدار . وسوف يحسب مقياس رسم للعرض البياني حتى يمكن رؤية جميع نقط البيانات وملء الشاشة بأكملها بصرف النظر عن قيم البيانات المعطاه . وسوف نستخدم أسلوب بيانات التحليل المتوسط لتوليد العرض .
أساسا فإن المسألة هي لتوفيق المعادلة الخطية .

$$y = ax + b$$

لمجموعة من نقط البيانات : (Y_1, X_1) ، (Y_2, X_2) ، (Y_m, X_m) بتصغير مربعات الاخطاء الى الحد الأقصى كما هو مبين في مثال ٧ - ٢٢ .

وتكون الكميات المجهولة هي قيم المعاملات a و b . وسنحصل على هذه القيم باستخدام الصيغ الرياضية التالية :—

$$a = (Md_2 - c_1d_1)/(Mc_2 - c_1^2)$$

$$b = (d_1 - ac_1)/M$$

where

$$c_1 = \sum_{i=1}^M x_i$$

$$c_2 = \sum_{i=1}^M x_i^2$$

$$d_1 = \sum_{i=1}^M y_i$$

$$d_2 = \sum_{i=1}^M x_i y_i$$

* في الحقيقة ان مصطلح الانحدار الخطي يطبق على توفيق منحنى يحوى عدة أنواع مختلفة من المنحنيات معتمدا دوال أسية ، دوال لوغاريتمية وكثيرات الحدود (أنظر مثال ٧ - ٢٢) .

وعلى ذلك فإن الطريقة الكلية تكون كما يلي :—

- ١ — قراءة فقط البيانات (Y_1, X_1) و $(Y_2, X_2), \dots, (Y_m, X_m)$ ثم توليد المجاميع التراكميه c_1, c_2, d_1, d_2 كما سبق تعريفها .
- ٢ — الحل للحصول على المعاملات المجهولة a و b باستخدام الصيغ الرياضيه السابقة الموضحه بعاليه .
- ٣ — تحديد اكبر وأصغر قيمه لكل من X, Y اعدادا للخطوة ٤ التاليه .
- ٤ — عمل مقياس رسم لنقط البيانات حتى تملأ مساحة عرض البيانات كامله .
- ٥ — توليد العرض الحقيقى للبيانات فى ثلاث مراحل .
(أ) رسم المحاور .
(ب) رسم نقط البيانات المفرده (بعد عمل مقياس رسم لها) .
(ج) رسم خط الانحدار الناتج .
- ٦ — طباعة معادلة الانحدار المحسوبة فى أعلى الشاشة فوق العرض البياني .

. ويبين شكل ١٢ — ٨ برنامج يسك الحقيقى . وتمسح السطور من 30 الى 110 الشاشة وتولد عنوان البرنامج وتعرف الصفوف المطلوبة وتعطى قيماً مبدئيه لمتغيرات . وتعطى السطور 150 الى 200 برنامج البيانات المدخله . لاحظ ان البرنامج يسمح بادخال عدد غير محدود فى نقط البيانات . وتستمر طريقة ادخال البيانات حتى يضغط المستخدم على مفتاح Return عند تلقيه $Y(I)$ فيدخل بذلك فى سلسلة حروف فارغة .

ويحدد السطران 250,240 الميل (a) والجزء المقطوع من محور y لخط الانحدار المطلوب . ثم يتم فرز بيانات النقط الفردية لتحديد القيمه العظمى والصغرى لكل من $Y(1)$ و $X(1)$ فى السطور 290 الى 350 . ويتم تحديد المقياس الحقيقى لبيانات النقط فى السطور 390 الى 420 . السطور من 430 الى 450 تولد مقياس النقط النهائية لحساب خط الانحدار .
ويتم العرض البياني الحقيقى فى السطور من 490 الى 590 .

ويستدعى سطر 490 بيانات التحليل المتوسط بخلفيه سوداء ولوحه الوان رقم 1 (الوان سماوى ، بنفسجى وأبيض) . ويتبع هذا أوامر مخرجات البيانات والتي يتم تجميعها فى ثلاث اقسام مختلفه :—

السطور 500 الى 520 تولد المحاور ، والسطور 540 الى 570 تتسبب فى رسم بيانات النقط المنفردة . والسطر 590 يولد رسم خط الانحدار المحسوب . (لاحظ أن بيانات كل نقطه تنحصر فى مثلث صغير بحيث تكون مرئيه بوضوح على الشاشة وعلى ذلك فإنه يتم استخدام كل من الجملتين (LINE , PSET) .

وفى النهايه يتم عرض المعادله لحساب خط الانحدار وذلك فى أعلى الشاشة . ويولد السطران 630 و 640 هذا العرض . (لاحظ أن ذلك هو عرض للنص رغم أنه قد تم توليده بأسلوب البيانات) .

```

10 '***** LINEAR REGRESSION WITH GRAPHICAL DISPLAY *****
20 '
30 KEY OFF: CLS
40 WIDTH 80: SCREEN 0: COLOR 7,0
50 DIM X(100),Y(100)
60 LOCATE 1,20: PRINT STRING$(40,"*")
70 LOCATE 2,20: PRINT "      LINEAR REGRESSION ROUTINE      *"
80 LOCATE 3,20: PRINT "*,TAB(59);*"
90 LOCATE 4,20: PRINT "* Press RETURN after last data point *"
100 LOCATE 5,20: PRINT STRING$(40,"*"): PRINT
110 I=0: C1=0: C2=0: D1=0: D2=0
120 '
130 '***** ENTER DATA AND GENERATE COEFFICIENTS *****
140 '
150 I=I+1
160 PRINT "Y(,I,") = "; INPUT "",ANS$
170 IF ANS$="" THEN M=I-1: GOTO 240 ELSE Y(I)=VAL(ANS$)
180 LOCATE C$RLIN-1,20: PRINT "X(,I,") = "; INPUT "",X(I)
190 C1=C1+X(I): C2=C2+X(I)^2: D1=D1+Y(I): D2=D2+X(I)*Y(I)
200 GOTO 150
210 '
220 '***** SOLVE FOR UNKNOWN CONSTANTS *****
230 '
240 A=(M*D2-C1*D1)/(M*C2-C1^2): IF ABS(A) < 1E-08 THEN A=0
250 B=(D1-A*C1)/M: IF ABS(B) < 1E-08 THEN B=0
260 '
270 '***** FIND LARGEST AND SMALLEST X AND Y *****
280 '
290 XMAX=-100000!: YMAX=-100000!: XMIN=100000!: YMIN=100000!
300 FOR I=1 TO M
310   IF X(I) > XMAX THEN XMAX = X(I)
320   IF Y(I) > YMAX THEN YMAX = Y(I)
330   IF X(I) < XMIN THEN XMIN = X(I)
340   IF Y(I) < YMIN THEN YMIN = Y(I)
350 NEXT I
360 '
370 '***** SCALE THE X'S AND Y'S *****
380 '
390 FOR I=1 TO M
400   X(I)=29+INT(280*(X(I)-XMIN)/(XMAX-XMIN))
410   Y(I)=189-INT(150*(Y(I)-YMIN)/(YMAX-YMIN))
420 NEXT I
430 X1=29: X2=309
440 Y1=189-INT(150*((A*XMIN+B)-YMIN)/(YMAX-YMIN))
450 Y2=189-INT(150*((A*XMAX+B)-YMIN)/(YMAX-YMIN))
460 '
470 '***** PLOT THE GRAPHICAL DISPLAY *****
480 '
490 SCREEN 1,1: COLOR 0,1
500 LINE (19,29)-(19,199): LINE -(319,199)
510 FOR IY=59 TO 179 STEP 20: LINE (19,IY)-(23,IY): NEXT IY
520 FOR IX=39 TO 299 STEP 20: LINE (IX,199)-(IX,195): NEXT IX
530 '
540 FOR I=1 TO M
550   PSET(X(I),Y(I)): LINE (X(I),Y(I)-2)-(X(I)-4,Y(I)+2)
560   LINE -(X(I)+4,Y(I)+2): LINE -(X(I),Y(I)-2)
570 NEXT I
580 '
590 LINE (X1,Y1)-(X2,Y2)
600 '
610 '***** PRINT THE REGRESSION EQUATION *****
620 '
630 LOCATE 1,5: PRINT "Y =";A;" X ";
640 IF B >= 0 THEN PRINT "+";B ELSE PRINT "-";ABS(B)
650 END

```

ويبين الشكلان ١٢ - ٩ و ١٢ - ١٠ مظهر الشاشة عند تنفيذ البرنامج . ويبين شكل ١٢ - ٩ مرحلة ادخال البيانات التي تحدث في أسلوب النص لمجموعة البيانات الآتية : (تم وضع خط تحت البيانات المدخلة في شكل ١٢ - ٩) .

i	y_i	x_i
1	225	10
2	287	20
3	429	30
4	542	40
5	587	50
6	744	60
7	831	70
8	880	80

لاحظ موضع المؤشر الومضي التالي للتلقيين (9) Y . ولما كان قد تم ادخال كل البيانات عند هذه النقطة ، فالمستخدم يقوم ببساطة بالضغط على مفتاح Return استجابة للتلقيين . وهذا ينهي مرحلة ادخال البيانات .

```
*****
*          LINEAR REGRESSION ROUTINE          *
*                                                                 *
*   Press RETURN after last data point   *
*****
```

Y(1) = <u>225</u>	X(1) = <u>10</u>
Y(2) = <u>287</u>	X(2) = <u>20</u>
Y(3) = <u>429</u>	X(3) = <u>30</u>
Y(4) = <u>542</u>	X(4) = <u>40</u>
Y(5) = <u>587</u>	X(5) = <u>50</u>
Y(6) = <u>744</u>	X(6) = <u>60</u>
Y(7) = <u>831</u>	X(7) = <u>70</u>
Y(8) = <u>880</u>	X(8) = <u>80</u>
Y(9) = -	

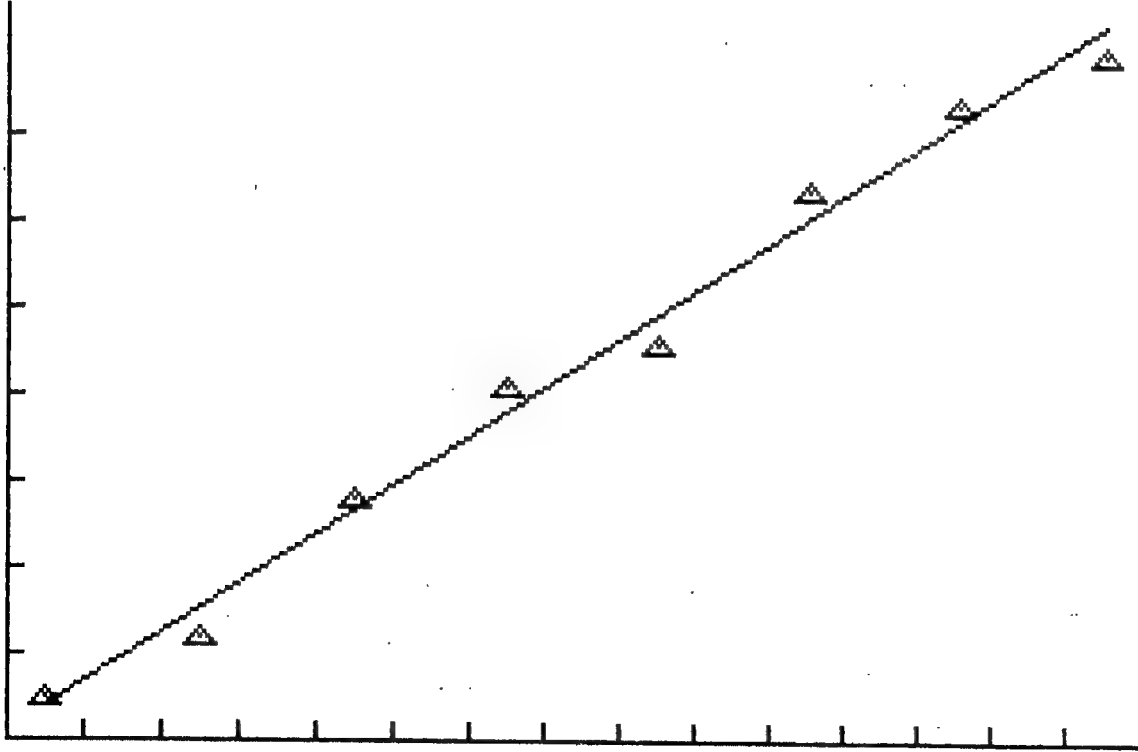
شكل ١٢ - ٩

ويبين شكل ١٢ - ١٠ (صفحة ٣٧٢) العرض البياني المناظر والمعادلة لحساب خط الانحدار هي :

$$y = 9.875x + 121.25$$

وتقوم هذه المعادلة بتوليد الخط المبين الذي يتضح أنه يمر بنقط البيانات المعطاه رغم أنه قد تم عرض الخط حسب المقياس ليملا الشاشة . ويبين العرض البياني الدقة التي يمثل بها خط الانحدار المحسوب لنقط البيانات .

$$Y = 9.875 * X + 121.25$$



شكل ١٢ - ١٠

١٢ - ٣ الأشكال SHAPES

تحتوى معظم نسخ بيسك الحاسب الدقيق على جمل تسمح برسم الأشكال البسيطة ، فمثلا يحتوى ميكروسوفت بيسك على صورة خاصة من جملة LINE التي تولد مستطيلات وجملة CIRCLE التي يمكن أن تولد دوائر وقطاعات ناقصة . ويمكن ملء هذه الأشياء بلون متاح إذا رغبتنا في ذلك . ويمكن تكوين تأثيرات بيانية مشوقة بتوفيق هذه الأشياء بطرق مختلفة .

افترض مثلاً جملة LINE التي سبق مناقشتها في القسم الأخير ، فإذا كانت الجملة تنتهى بالحرف B أى : LINE (20,50) - (300,150),3,B فإنه يمكن تفسير زوج الإحداثيات على أنهما الركنان المقابلان للمستطيل (أى الصندوق) وعلى ذلك فإنه يمكن رسم المستطيل الذى يكون قطره هو الخط الواصل بين هاتين النقطتين وذلك باللون المبين .

مثال ١٢ - ١٣

يحتوى برنامج ميكروسوفت بيسك مكتوب لحاسب IBM الشخصى على الجمل التالية :

```
10 SCREEN 1 : COLOR 4,1
20 LINE (20,50) - (300,150),3,B
```

سطر 10 يحدد بيانات التحليل المتوسط بخلفية حمراء ولوحة ألوان رقم 1 (ألوان : سماوى ، وبفسجى وأبيض) . يولد سطر 20 حدود المستطيل الأبيض الذى يصل قطره بين النقطتين (20,50) و (300,150) على خلفية حمراء . (قارن مع مثال ١٢ - ٩) .

وبين المثال التالى برنامج بيسك كاملاً يتم فيه توليد مستطيلات مستخدمة بطريقة أكثر إبداعاً .

مثال ١٢ - ١٤ المستطيلات المتمددة Expanding Rectangles

يبين شكل ١٢ - ١١ برنامج ميكروسوفت بيسك كاملاً مكتوباً لحاسب IBM الشخصى الذى يتسبب فى تحريك سلسلة من المستطيلات من مركز الشاشة إلى الحواف الخارجية . ويتم توليد مجموعات من المستطيلات بألوان متغيرة للواجهات الأمامية والخلفية مكونة بذلك خداعاً من نبضات لأشكال مستطيلة تنشأ عند مركز الشاشة . ويستخدم البرنامج أيضاً الصوت لتحسين التأثيرات البصرية .

```

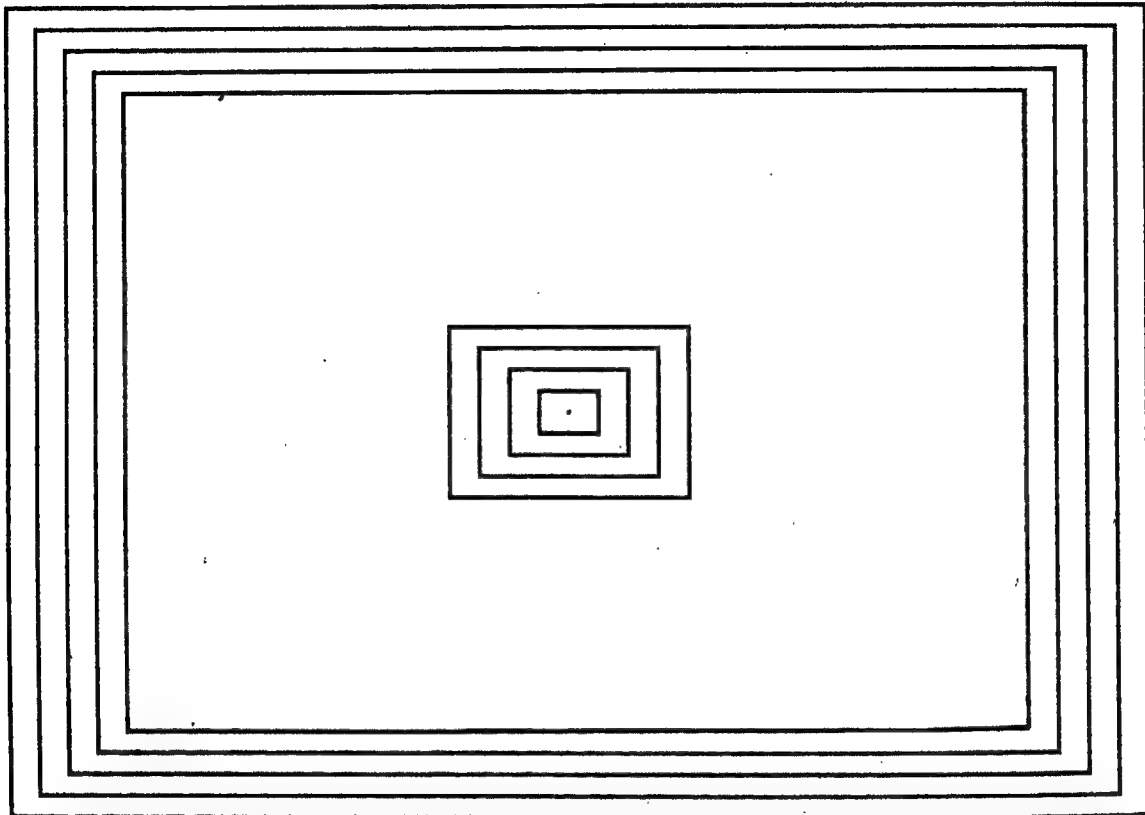
10 ***** EXPANDING RECTANGLES. *****
20
30 KEY OFF : CLS
40 SCREEN 1 : COLOR 0,1 : C=3
50 P=0 : Q=-45
60
70 *** BEGIN MAIN LOOP ***
80
90 LINE (160-1.6*P,100-P)-(160+1.6*P,100+P),C,B
100 IF Q >= 0 THEN LINE (160-1.6*Q,100-Q)-(160+1.6*Q,100+Q),0,B
110 P=P+5 : IF P > 95 THEN P=0 : C=1+INT(3*RND)
120 Q=Q+5 : IF Q > 95 THEN Q=0
130 IF P=0 OR Q=0 THEN SOUND 20+INT(1000*RND),10
140 GOTO 90
150 END

```

شكل ١٢ - ١١

ويقوم سطر 30 فى البرنامج بمسح الشاشة . ويستدعى سطر 40 بيانات التحليل المتوسط بخلفية سوداء ولوحة ألوان رقم 1 (ألوان : سماوى ، وبنفسجى ، وابيض) ، وبلون واجهة 3 (أبيض) . ويستخدم البرنامج عدادين Q,P يبدأان فى سطر 50 .

وتبدأ الحلقة التكرارية الأساسية بسطر 90 الذى يولد مستطيلاً يتحدد حجمه بالقيمة المخصصة إلى P (بداية P=0 مسيبه ظهور المستطيل الأول كنقطة عند مركز الشاشة) . ويتحدد لون المستطيل بالقيمة المعينة إلى C (أبيض فى البداية) .



شكل ١٢ - ١٢

ويقوم سطر 100 ايضا بتوليد مستطيل بشرط الا تكون Q سالبه . (وتأخذ Q في البدايه قيمه سالبه حتى يمكن أن ينشأ تأخير وتزداد بعد ذلك وتظل غير سالبه) . ومع كل ، فيتم توليد المستطيل بلون الخلفيه (اسود) .

ويتحدد حجم المستطيل بالقيمه المحدده Q . حيث أن P ، Q سوف تختلفان دائماً في القيمه ، فإن هذا المستطيل لن ينطبق على المستطيل الذى تم توليده في سطر 90 . وسيكون التأثير هو مسح أحد هذه المستطيلات المرسومة سابقا .

ويعوم السطران 110 ، 120 بزيادة P و Q على الترتيب لإعادة وضع كل معلمه إلى الصفر إذا كانت قيمتها تزيد عن 95 . ويتم أيضا توليد لون واجهة جديد عشوائيا عندما يعاد وضع P إلى الصفر .

ويولد سطر 130 صوتا عشوائيا عندما تبدأ سلسلة جديدة من المستطيلات (عندما يعاد وضع Q,P إلى الصفر) ثم يعيد سطر 140 التحكم إلى سطر 90 لمروور آخر خلال الحلقة التكرارية . لاحظ أن الحلقة سوف تستمر في التنفيذ بغير حدود .

وبين شكل ١٢ - ١٢ نوع العرض البياني الذى تم توليده بمجرد بدء تنفيذ البرنامج . ومع كل .. فكما هو الحال مع البرامج الأخرى من هذا النوع فإننا نشجع القارئ على تنفيذ هذا البرنامج فعلا حتى يمكنه تفهم التأثيرات الديناميكية التى تنشأ بالكامل .

نفرض الآن ان جمله LINE تنتهى بالحرفين BF بدلا من الحرف B ، أى 3,BF (300,150) - I.LINE (20,50) . ويمثل الحرف الاخير املا إلى «FILL» . وتسبب هذه المعلمة المستطيل (الذى تم توليده بالمعلمة B) بأن يملأ باللون المعين للواجهة .

مثال ١٢ - ١٥

افتراض مرة أخرى برنامج ميكروسوفت بيسك مكتوباً لحاسب IBM الشخصى الذى يحتوى على الجمل الآتية

```
10 SCREEN 1 : COLOR 0,0
20 LINE (20,50)-(300,150),1,BF
```

سطر 10 يستدعى بيانات التحليل المتوسط بخلفيه سوداء ولونه رقم 0 (الوان اخضر ، وبنى) . سطر 20 يتسبب في توليد مستطيل أخضر مصمت ، يصل قطره بين النقط (20 ، 20) و (300 ، 125) . ويتم عرض هذا المستطيل على خلفيه سوداء (قارن مع مثالى ١٢ - ٩ و ١٢ - ١٣) .

مثال ١٢ - ١٦ *المشكال A Kaleidoscope

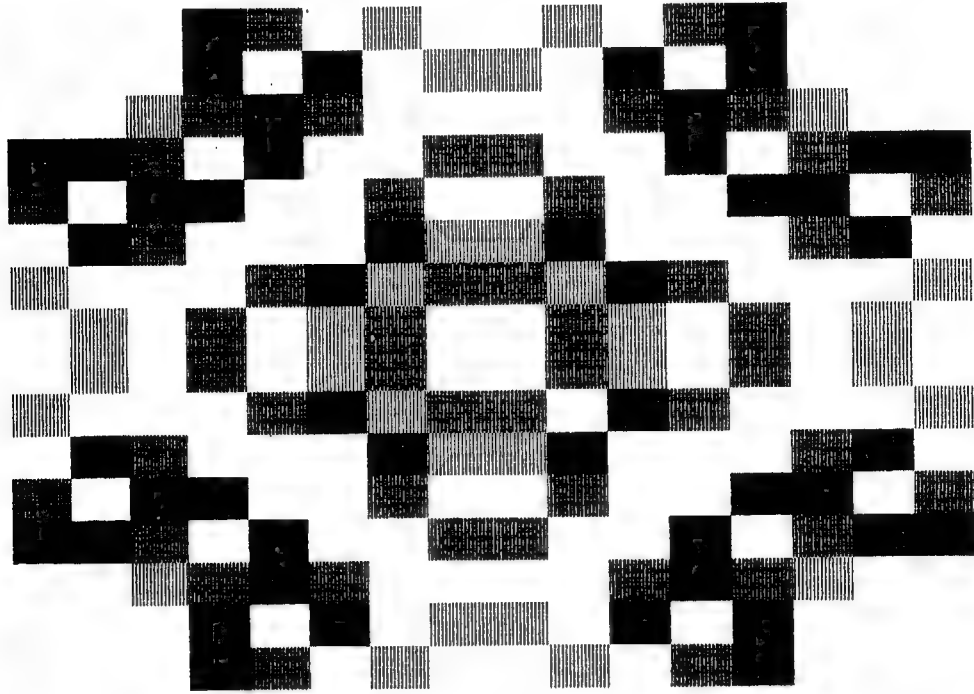
أحد التطبيقات المشوقة لاستخدام المستطيلات المليئة هو محاكاة المشكال . وهذا يتضمن توليد قطاعات ملونة وعشوائيه صغيرة (أى مستطيلات) عند أوضاع عشوائيه في وتد يشغل $\frac{1}{8}$ الشاشة . وتكرر كل كتلة في السبع أوتاد الأخرى عند أوضاع متماثلة بالنسبة إلى الوضع

```
10 '***** KALEIDOSCOPE *****
20 '
30 KEY OFF : CLS : SCREEN 1
40 '
50 '*** BEGIN MAIN LOOP ***
60 '
70 U=INT(10*RND) : X=9-INT((U+1)*RND) : Y=9-INT((U+1)*RND) : C=INT(4*RND)
80 LINE (16*X,10*Y)-(16*X+15,10*Y+9),C,BF
90 LINE (304-16*X,10*Y)-(319-16*X,10*Y+9),C,BF
100 LINE (16*X,190-10*Y)-(16*X+15,199-10*Y),C,BF
110 LINE (304-16*X,190-10*Y)-(319-16*X,199-10*Y),C,BF
120 LINE (16*Y,10*X)-(16*Y+15,10*X+9),C,BF
130 LINE (16*Y,190-10*X)-(16*Y+15,199-10*X),C,BF
140 LINE (304-16*Y,10*X)-(319-16*Y,10*X+9),C,BF
150 LINE (304-16*Y,190-10*X)-(319-16*Y,199-10*X),C,BF
160 GOTO 70
170 END
```

شكل ١٢ - ١٣

* أداه تحتوى على قطع متحركة من الزجاج الملون وما ان تتغير أوضاعها حتى تعكس مجموعة لانهائية لها من الأشكال الهندسية المختلفة الألوان .

الأصلى . ويمكن جعل المحاكاة أكثر تشويقاً بإدخال تحيز ، بحيث يتم توليد معظم الكتل أقرب لمركز الشاشة .
ويحتوى شكل ١٢ - ١٣ على البرنامج الفعلى . ويمسح سطر 30 الشاشة ويستدعى بيانات التحليل المتوسط . وتكون الخلفية السوداء (لون 0) واللوحة رقم 0 (ألوان : أخضر ، وأحمر ، وبنى) نشيطتين لغياب البديل الافتراضى .
ويحدث التوليد المتكرر لقطاعات الألوان العشوائية فى الحلقة التكرارية الرئيسية التى تحتوى على السطور 70 إلى 160 . ويولد كل مرور لهذه الحلقة التكرارية ثمانية قطاعات يكون وضعها متماثلاً بالنسبة لكل واحدة منها ، مع وجود كتلة واحدة فى كل وتد . ويكون لكل مجموعة من الكتل نفس اللون ، فعلى ذلك .. فسطر 70 ينشئ وضع الكتلة الأولى ولون كل الكتل التالى . لاحظ أن قيم الإحداثيين Y,X تكون معتمدة على المعلمة U التى يتم توليدها عشوائياً . ويتسبب هذا فى تجمع أزواج الإحداثيات تجاه مركز الشاشة .
وكل من السطور 80 إلى 150 تولد كتلة واحدة . ويكون عرض كل كتلة 16 نقطة «Pixels» وارتفاعها 10 نقط (10 خطوط ممسوحة) .
ويم اختيار النقط النهائية بطريقه تؤدي إلى أن تكون الكتل نموذجاً متماثلاً على الشاشة . وهذا التماثل هو الذى يحاكى عمل المشكال .
يبين شكل ١٢ - ١٤ نمطاً نموذجياً . ومع كل .. نشجع القارئ على أن ينفذ البرنامج فعليا حتى يتفهم ديناميكية النموذج الذى يتكون .



شكل ١٢ - ١٤

تحتوى بعض نسخ ميكرو سوفت بيسك على جملة CIRCLE التى تسمح برسم النوائر والأقواس والقطاعات الناقصة . وهذه الجملة فى أبسط صورها تحتوى على كلمة CIRCLE يعقبها زوج من الإحداثيات محصور بين قوسين وتفصله فاصله . ولا بد أن يعقب الإحداثيات قيمة لنصف القطر أى 80, (160,100) CIRCLE .

وقد يظهر رقم صحيح بين اختيار اللون بعد نصف القطر كشيء اختياري أى 3, 80, (160,100) CIRCLE . فإذا لم يتحدد اللون بهذه الطريقة فإن لون رقم 3 سيتم اختياره تلقائياً من اللوحة الحالية

مثال ١٢ - ١٧

تحتوى الجمل الآتية على برنامج ميكروسوفت بيسك متقدم (BASICA)، مكتوب لحاسب IBM الشخصى .

10 SCREEN 1 : COLOR 4,1
20 CIRCLE (160,100),80,3

يحدد سطر 10 بيانات التحليل المتوسط بخلفية حمراء ولوحة ألوان رقم 1 (الألوان : سماوى وبنفسجى وأبيض) . يولد سطر 20 محيط دائرة بيضاء مركزها عند منتصف الشاشة ، أى النقطة (100 و 160) ونصف قطرها 80 نقطة .

مثال ١٢ - ١٨ الدوائر المتعددة Expanding Circles

يحتوى شكل ١٢ - ١٥ على برنامج ميكروسوفت بيسك متقدم ، مكتوب لحاسب IBM الشخصى . يتسبب هذا البرنامج فى توليد مجموعات متكررة من دوائر مركزية (متحدة المركز) ، وتبدأ كل مجموعة منها عند مركز الشاشة وتحرك قطريا للخارج باتجاه الحافة . سوف ترسم كل مجموعة من الدوائر بلون مفرد يختار عشوائيا . ومع توليد مجموعة متتالية من الدوائر ، فإن الدوائر الجديدة سوف تنشئ نماذج تتداخل مع الدوائر السابقة ، مكونة بذلك نماذج شيقة وملونة .

```
10 '***** CIRCLES *****
20 '
30 KEY OFF : CLS
40 SCREEN 1 : COLOR 0,1
50 LINE (39,0)-(279,199),3,B
60 CLR=1
70 '
80 '*** BEGIN LOOP ***
90 '
100 FOR R=5 TO 120 STEP 5
110   CIRCLE (160,100),R,CLR
120 NEXT R
130 CLR=CLR+1 : IF CLR > 3 THEN CLR=1
140 GOTO 100
150 END
```

شكل ١٢ - ١٥

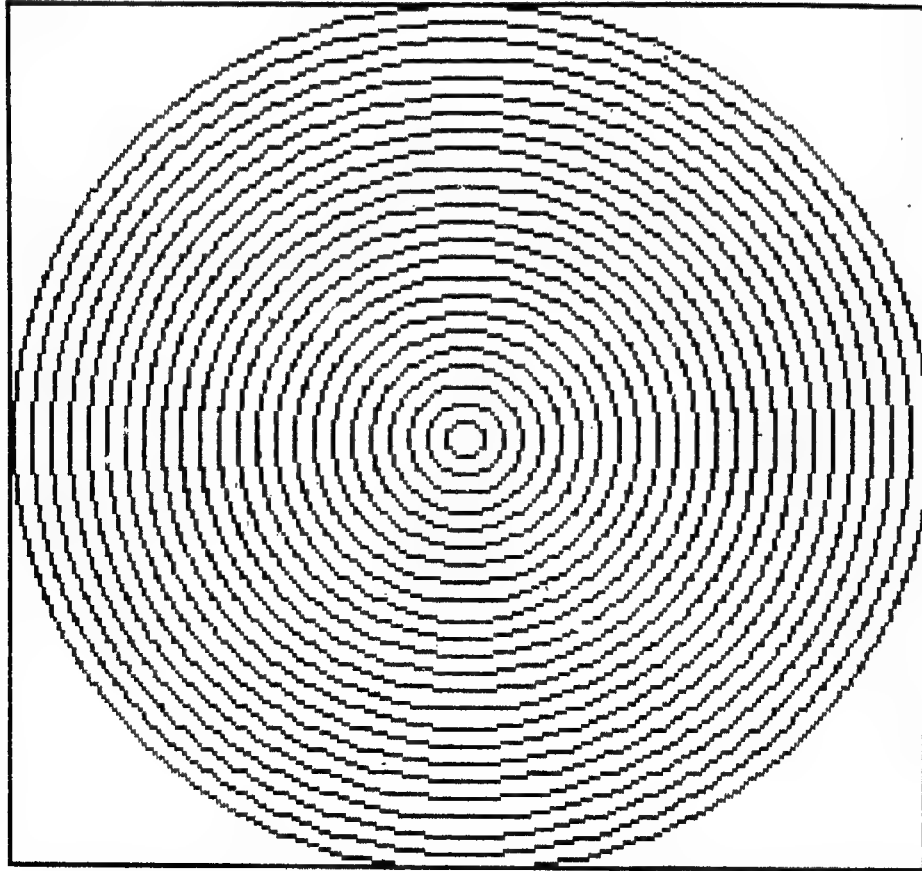
وسنعتبر هذا البرنامج بالتفصيل . ويلغى سطر 30 أى عرض سابق لتعريفات مفتاح الدالة وبموجب الشاشة ، بينما يحدد سطر 40 بيانات التحليل المتوسط ، بخلفية سوداء ولوحة ألوان رقم 1 (ألوان : سماوى ، بنفسجى وأبيض) . يولد السطر 50 مربعا أبيض حول الحواف الخارجية للشاشة . ويتم تحديد لون سماوى لأول مجموعة من الدوائر فى سطر 60 .

ويبدأ الجزء المتكرر من البرنامج بسطر 100 ويستمر حتى سطر 140 . وتكون السطور 100 و 120 الحلقة التكرارية FOR—TO التى تولد مجموعة من الدوائر المركزية بتحدد لونها بالمعلم CLR . ثم تتغير قيمة معلمة اللون عندئذ فى سطر 130 . (لاحظ أن قيمة هذه المعلمة سوف تظل دائما 1 أو 2 أو 3) . ويعيد سطر 140 التحكم إلى الحلقة التكرارية FOR—TO مولداً بذلك مجموعة أخرى من الدوائر .. وهكذا .

يبين شكل ١٢ - ١٦ نوع النماذج التى قد يولدها هذا البرنامج . ومع كل فلابد أن تفهم أن هذه النماذج تحتوى على اختلافات مشوقة فى اللون ، وأنها فى حركة ثابتة ، وعلى ذلك يجب أن يلاحظ القارئ البرنامج أثناء التنفيذ حتى يتفهم ماذا يحدث .

ولا تحتوى جملة CIRCLE على ما يؤدى إلى ملء الدائرة بأى لون غير لون الخلفية . ومع كل .. فإن معظم نسخ ميكروسوفت بيسك التى تحتوى جملة CIRCLE تحتوى أيضاً على جملة PAINT ، والتى تسمح لأى شكل مغلق (بما فيه الدوائر والقطاعات الناقصة والمستطيلات) أن يتم ملؤه بألوان أخرى . تتكون هذه الجملة من الكلمة PAINT يتبعها زوج من الإحداثيات بين قوسين ويفصلهما فاصلة أى PAINT (160,100) . ويمكن للإحداثيين أن يمثلأ أى نقطة تنحصر داخل الشكل الذى يراد ملؤه .

ويمكن أن يتبع الإحداثيان رقماً صحيحاً ليحدد اللون أى 3, PAINT (160,100) وإذا لم يتم تحديد اللون فإنه سيتم اختيار اللون رقم 3 تلقائياً من اللوحة الحالية .



شكل ١٢ - ١٦

مثال ١٢ - ١٩ السهم الممتلئ المضيء **A Filled Lightning Bolt**

دعنا مرة أخرى نفحص البرنامج المعطى في مثال ١٢ - ١٠ الذى يولد السهم المضيء (انظر شكل ١٢ - ٤) . افترض أننا الآن أضفنا جملة PAINT لهذا البرنامج أى :

90 PAINT (100,40),2

يبين شكل ١٢ - ١٧ البرنامج الكامل . لاحظ أن الإحداثيات (100,40) التى تحدد بجملة PAINT تشير إلى النقطة المحصورة داخل الشكل . لاحظ أيضاً أن اللون (2) هو نفس ما تحدده جملة LINE . وعلى ذلك تأثير جملة PAINT هو توليد شكل مصمت أحمر .

```

10 '***** LIGHTNING BOLT *****'
20 '
30 KEY OFF : CLS
40 SCREEN 1 : COLOR 0,0
50 LINE (20,20)-(120,80),2 : LINE -(80,80),2
60 LINE -(220,180),2 : LINE -(140,100),2
70 LINE -(190,100),2 : LINE -(110,20),2
80 LINE -(20,20),2
90 PAINT (100,40),2
100 END

```

شكل ١٢ - ١٧

ويوضح شكل ١٢ - ١٨ الشكل الذى قد تم توليده بواسطة هذا البرنامج (قارن مع شكل ١٢ - ٥) . تذكر أن الشكل سوف يظهر بالأحمر أمام خلفية سوداء إذا تم تنفيذ البرنامج على حاسب ذى شاشة ملونة .

ويمكن أن يتبع معلمة اللون فى جملة PAINT معلمة إضافية تمثل لون الحدود الجانبية ، أى 2,3,(160,100) PAINT ولا بد للشكل المطلوب ملؤه أن يتفق مع هذا اللون . وعلى ذلك إذا كانت النقطة التى تحدد بجملة PAINT تكون داخل أكثر من شكل واحد وكل شكل له لون مختلف فإن لون الحدود الجانبية سوف يحدد الشكل المطلوب ملؤه .



شكل ١٢ - ١٨

مثال ١٢ - ٢٠

نفرض أن برنامج ميكروسوفت متقدماً يحتوى على الجمل الأربع الآتية :

```
10 SCREEN 1 : COLOR 4,1
20 CIRCLE (160,100),80,2
30 CIRCLE (160,100),60,3
40 PAINT (160,100),3,2
```

وتسبب هذه الجمل توليد دائرتين مركزيتين . وستكون الدائرة الخارجية (معرفة بسطر 20) بنفسجية بنصف قطر 80 . وستكون الدائرة الداخلية (سطر 30) بيضاء بنصف قطر 60 . وتسبب جملة PAINT فى سطر 40 بأن تملأ الدائرة الخارجية بلون رقم 3 (أبيض) حيث أن لون الحدود فى سطر 40 (أى لون رقم 2) ينطبق على اللون الذى يتحدد فى سطر 20 .

ويمكن استخدام جملة CIRCLE ليس فقط لتوليد الدوائر كاملة ولكن أيضاً أجزاء من الدوائر (أقواس) ولإجراء ذلك فإنه لا بد أن يتبع معلمة اللون معلمتين إضافيتين : زاوية البداية وزاوية النهاية . وتقاس هاتان الزاويتان بالطريقة الهندسية التقليدية ، وتزايد فى اتجاه عكس عقرب الساعة فى النصف الأيمن للمحور الأفقى (أى النصف الأيمن لمحور السينات) . ولا بد من التعبير عن كل من الزاويتين بالتقدير الدائرى ، وعلى ذلك فلا بد أن تقع فى مدى من 0 إلى 2π أى 0,1,0,3,14 CIRCLE (160,100),80,1,0,3,14 .

وإذا كانت جملة CIRCLE لا تحتوى على قيمة معلمة اللون ولكنها تحتوى على زاويتي البداية والنهاية ، فإن قى نصف القطر لابد أن يتبعها فاصلتين أى CIRCLE (160, 100), 80,0,3.14 . وتعتبر الفاصلتين المتتاليتين عن معمة لون خالية .

مثال ١٢ - ٢١

يحتوى برنامج ميكروسوفت بيسك متقدم (BASICA) مكتوب لحاسب IBM الشخصى على الجمل الآتية :

```
10 SCREEN 1 : COLOR 0,0
20 CIRCLE (160,100),80,1,0,3.14
```

يعين سطر 10 بيانات التحليل المتوسط بخلفية سوداء ولوحة ألوان رقم 0 (ألوان أخضر ، أحمر وبنى) . ثم يولد سطر 20 النصف الأعلى لدائرة خضراء بنصف قطر 80 . (لاحظ أن زاويتي البداية والنهاية هما π و 0 على الترتيب . إذا استبدلنا سطر 20 مع

```
20 CIRCLE (160,100),80,1,3.14,0
```

فإن النصف الأسفل من دائرة خضراء سوف يتم توليده حيث أن زاويتي البداية والنهاية هما 2π و π على الترتيب . نفس هذا التأثير سوف يحدث بكتابة .

```
20 CIRCLE (160,100),80,1,3.14,6.28
```

الآن افترض أن النسخة العادية من سطر 20 قد تم تبديلها مع

```
20 CIRCLE (160,100),80,,0,3.14
```

سوف تولد هذه الجملة النصف الأعلى لدائرة إلا أنها ستكون باللون البنى بدلاً من الأخضر (حيث أن قيمة معلمة اللون لم تحدد صراحة فإن القيمة البديية 3 سوف يتم تطبيقها فيظهر اللون البنى) .

يمكن أن تكون زاويتي البداية والنهاية فى جملة CIRCLE سالبة كما يمكن أن تكون موجبة أى CIRCLE (160, 100), 80,1,-3.14,-6.28 . وسيظل تفسير الزوايا السالبة كما لو كانت موجبة ، بالمفهوم الهندسى . ومع كل فإن تأثير الزاوية السالبة هو توصيل النقطة النهائية المناظرة بمركز الدائرة .

مثال ١٢ - ٢٢

افترض الآن برنامج ميكروسوفت بيسك متقدم والموجود فى المثال الأخير يحتوى على الجمل الآتية :-

```
10 SCREEN 1 : COLOR 0,0
20 CIRCLE (160,100),80,1,-3.14,-6.28
```

سوف يولد سطر 20 النصف الأسفل لدائرة خضراء مع خط أفقى يصل النقطتين النهائيتين .

مثال ١٢ - ٢٣

الجمل الآتية سوف تولد شكل مألوف لعدد من ألعاب الفيديو البحرية .

```
10 SCREEN 1 : COLOR 0,1
20 PI=3.141593
30 CIRCLE (160,100),40,2,-PI/4,-2*PI
40 CIRCLE (170,78),5,0
50 PAINT (150,100),2,2
```

لاحظ أن الشكل سوف يمتلئ بالبنفسجى .

مثال ١٢ - ٢٤ مولد خريطة دائرية A Piechart Generator

الخريطة الدائرية هي شكل دائري يستخدم عادة لتمثيل البيانات في شكل نسب مئوية . ويتم تمثيل كل جزء من المعلومات (أى كل نسبة مئوية) بدلالة قطاع دائري . يتناسب محيط كل قطاع مع قيمة نقط البيانات المناظرة . وهكذا القيمة ٤٠ في المائة سوف يمثلها قطاع دائري محيطه هو ٤٠ في المائة من الدائرة الكلية .

ويبين شكل ١٢ - ١٩ مولد بيسك كاملاً لخريطة دائرية ، مكتوباً بميكروسوفت بيسك متقدم لحاسب IBM الشخصي . ومن المفروض أن كل قطعة من البيانات تحتوى على عنوان وقيمة عددية للنسبة المئوية . وعلى ذلك..يصاحب كل قطاع داخل الخريطة الدائرية عنوان مناسب .

```

10 ***** PIECHART GENERATOR *****
20
30 KEY OFF: CLS
40 WIDTH 80: SCREEN 0: COLOR 7,0
50 DIM TITLE$(6),PERCENT(6),A(6)
60 LOCATE 1,20: PRINT STRING$(40,"*")
70 LOCATE 2,20: PRINT " ";TAB(59);"*"
80 LOCATE 3,20: PRINT "          PIECHART GENERATOR          *"
90 LOCATE 4,20: PRINT " ";TAB(59);"*"
100 LOCATE 5,20: PRINT STRING$(40,"*"): PRINT
110
120 ***** ENTER DATA FOR EACH SECTOR *****
130
140 LOCATE 8,1: INPUT "Enter number of sectors (1-6): ",N: PRINT
150 IF N < 1 OR N > 6 THEN BEEP: LOCATE 8,32: PRINT SPACE$(6): GOTO 140
160 SUM=0
170 FOR I=1 TO N
180   PRINT "Sector";I;SPC(8);
190   INPUT "Title: ",TITLE$(I)
200   LOCATE C$RLIN-1,40: INPUT "Percent: ",PERCENT(I)
210   SUM=SUM+PERCENT(I)
220 NEXT I
230 IF SUM < 99.9 OR SUM > 100.1 THEN BEEP: CLS: LOCATE 23:
PRINT "Percentages do not sum to 100 - Please try again": GOTO 60
240
250 ***** GENERATE THE PIECHART *****
260
270 SCREEN 2
280 PI=3.14
290 A1=0
300 FOR I=1 TO N
310   A2=A1+2*PI*PERCENT(I)/100
320   A(I)=(A1+A2)/2
330   CIRCLE (320,100),150,1,-A1,-A2
340   A1=A2
350 NEXT I
360
370 ***** LABEL THE SECTORS *****
380
390 FOR I=1 TO N
400   C1=(320+75*COS(A(I)))\8+3
410   IF A(I) > PI/2 AND A(I) < 3*PI/2 THEN C1=C1-6
420   R1=(100-30*SIN(A(I)))\8+1
430   IF A(I) < PI THEN R1=R1-1 ELSE R1=R1+1
440   LOCATE R1,C1: PRINT PERCENT(I);"%"
450
460   C2=(320+150*COS(A(I)))\8+5
470   IF A(I) > PI/2 AND A(I) < 3*PI/2 THEN C2=C2-LEN(TITLE$(I))-8
480   R2=(100-62.5*SIN(A(I)))\8+1
490   IF A(I) < PI THEN R2=R2-1 ELSE R2=R2+1
500   LOCATE R2,C2: PRINT TITLE$(I);
510 NEXT I
520 END

```

وسوف يولد هذا البرنامج الخريطة الدائرية التي تحتوي على ستة قطاعات على الأكثر . ويحتوى البرنامج على أربعة أقسام رئيسية : قسم العنوان والبيانات وقسم ادخال البيانات وقسم بيانات التحليل العالى الذى يولد الخريطة الدائرية الفعلية وقسم الختام الذى يعنون كل من القطاعات . دعنا نخص التعليمات المفردة ببعض التفصيل . يحدد سطر 30 أى تعريفات لمفاتيح الدالة السابقة ويمسح الشاشة . يحدد سطر 40 أسلوب النص . بحروف بيضاء على خلفية سوداء . ويتم تعريف الصفوف المطلوبة للبرنامج فى سطر 50 . ويتم توليد عرض ابتدائى للشاشة فى السطور 60 وحتى 100 .

وتقوم السطور 140 حتى 230 بتكوين قسم إدخال البيانات . ويلقن سطر 140 عدد القطاعات . ولا بد أن تكون هذه القيمة أكبر من الصفر ولكنها لا يمكن أن تزيد عن ستة . يجرى سطر 150 اختبار خطأ للعدد المعين للقطاعات ويعود إلى السطر 140 إذا كانت القيمة المدخلة خارج المدى (حاول ثانية) . ويبدأ المجموع التراكمى لكل النسب فى سطر 160 . وتلقن السطور 170 حتى 220 لكل بنود البيانات (أولاً عنوان كل قطاع وبعدئذ النسبة المئوية المناظرة) . ويتم تجميع النسب المئوية كما أدخلت . وفى النهاية يختبر سطر 230 ما إذا كان مجموع النسب المئوية كلها يساوى 100 (أو إقريباً منها جداً) . وإذا لم يكن كذلك تم إعادة توليد العرض الابتدائى للشاشة وتم إعادة جزء البرنامج الخاص لكل البيانات المدخلة .

ويتم توليد الخريطة الدائرية الفعلية فى سطور 270 إلى 350 . ويتسبب سطر 270 فى تحويل البرنامج إلى بيانات التحليل العالى ويمسح الشاشة فى هذه العملية . ويحدد السطران 280 و 290 القيم العددية للمتغيرات A1,PI (زاوية البداية للقطاع الأول) . ثم تولد السطور 300 إلى 350 القطاعات المفردة للخريطة الدائرية . وستكون زاوية البداية معروفة دائماً . وعلى ذلك فإنه يتم حساب زاوية النهاية فى سطر 310 ويتم تحديد متوسط زاوية ممثلة فى سطر 320 . ويتم تخزين هذه القيمة الأخيرة فى كتلة متزامنة لاستخدامها فيما بعد عند وضع عنوان الخريطة الدائرية . ثم يتم توليد القطاع نفسه فى سطر 330 (لاحظ استخدام جملة CIRCLE بقيم بداية ونهاية سالبة) . وفى النهاية يتم تعيين قيمة زاوية البداية فى سطر 340 للأعداد للقطاع التالى .

تسبب السطور 390 إلى 510 فى عنوان الخريطة الدائرية . لاحظ أن ذلك يتم أثناء بقاء البرنامج فى أسلوب بيانات التحليل العالى . ويتحدد موضع كل نسبة مئوية فى السطور 400 إلى 440 باستخدام متوسط الزوايا المحسوبة فى القسم السابق ، ثم يتم عرض النسب المئوية (ولا داعى لمناقشة تفاصيل هذه الجمل المفردة إلا لبيان أن خاصية كل نص لها ارتفاع من 8 نقط ، ولهذا نحتاج إلى عدد صحيح يقبل القسمة على 8) . وبالمثل يتم تحديد موضع كل عنوان ثم يتم عرضه فى سطور 460 إلى 500 . وبذلك فإنه يتم توليد الخريطة الدائرية كاملة بعنوانين مناسبة وذلك مع تكملة هذه الحلقة التكرارية . نفرض أن البرنامج المنفذ يستخدم مجموعة البيانات المدخلة التالية . وبين شكل ١٢ — ٢٠ دياالج المدخلات (لاحظ أن استجابة المستخدم تحتها خط) . وبين شكل ١٢ — ٢١ الخريطة الدائرية الناتجة .

Source of Revenue	Percentage
Tuition	45
State aid	25
Research	15
Gifts	8
Other	7

وهناك معلمة واحدة إضافية تصاحب جملة CIRCLE لم تتم مناقشتها وهى معلمة المظهر . وتستخدم لتكون قطاعات ناقصة وأقواس قطاعات ناقصة بدلاً من الدوائر والأقواس الدائرية .

لا بد أن تتبع معلمة المظهر زاويتى البداية والنهاية فى جملة CIRCLE أى 2,14,3,1,0,80,100,160) CIRCLE . ولا بد أن تكون عدداً موجباً (وليس بالضرورة عدداً صحيحاً) أو ما ينتج عنه قيمة عددية موجبة . والقيمة 1 ينتج عنها شكل دائرى أو قريب من ذلك (وقد تختلف القيمة اللازمة للحصول على دائرة تماماً بعض الشيء من حاسب لآخر) . وإذا كانت القيمة أقل من 1 فإنها تولد قطع ناقص أفقى بينما تولد قيمة أكبر من 1 قطع ناقص رأسى . وكلما اختلفت معلمة المظهر عن 1 زاد الاختلاف المركزى .

وعندما تستخدم جملة CIRCLE لتوليد قطع ناقص أو قوس ناقص تشير معلمة نصف القطر إلى طول المحور الأكبر .

```

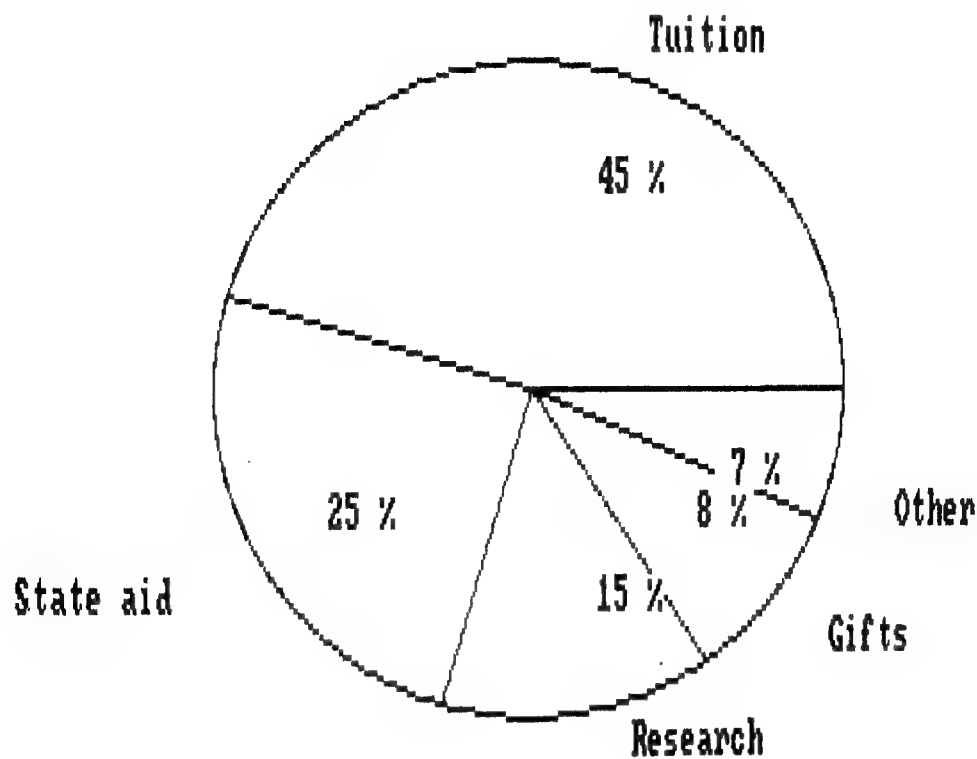
*****
*                                     *
*                               PIECHART GENERATOR                               *
*                                     *
*****

```

Enter number of sectors (1-6): 5

Sector 1	Title: <u>Tuition</u>	Percent: <u>45</u>
Sector 2	Title: <u>State aid</u>	Percent: <u>25</u>
Sector 3	Title: <u>Research</u>	Percent: <u>15</u>
Sector 4	Title: <u>Gifts</u>	Percent: <u>8</u>
Sector 5	Title: <u>Other</u>	Percent: <u>7</u>

شكل ١٢ - ٢٠



شكل ١٢ - ٢١

مثال ١٢ - ٢٥

يتضمن برنامج ميكروسوفت بيسك متقدم (BASICA) مكتوب لحاسب IBM الشخصي .

```

10 SCREEN 1 : COLOR 0,0
20 PI=3.141593
30 CIRCLE (160,100),80,1,0,PI,.5

```


سطر 10 يحدد بيانات التحليل المتوسط بخلفية سوداء ولوحة ألوان رقم 0 (ألوان أخضر ، أحمر وبني) . ويخصص سطر 20 قيمة للمعلمة PI . ويولد سطر 30 قوس قطع ناقص أخضر يتراوح من 0 إلى π . لاحظ أن القيمة للمعلمة المظهر أقل من 1 . وعلى ذلك فإن المحور الأكبر لقوس القطع الناقص يكون أفقياً . هذا بالإضافة إلى أن طول المحور الأكبر هو 160 نقطة حيث أنه قد تم تخصيص القيمة 80 لنصف القطر .

مثال ١٢ - ٢٦

نفرض الآن أن برنامج ميكروسوفت بيسك متقدم يحتوى على الجمل الآتية :-

```
10 SCREEN 1 : COLOR 0,1
20 CIRCLE (160,100),50,,,,,5
```

يحدد سطر 10 بيانات التحليل المتوسط بخلفية سوداء ولوحة ألوان رقم 1 (ألوان سماوى ، بنفسجى وأبيض) . و سطر 20 يتسبب في توليد قطع ناقص أفقى أبيض عند مركز الشاشة (لاحظ أن اللون الأبيض هو اللون البديى المناظر لقيمة 3) . ويكون طول المحور الأكبر هو 100 وحدة حيث أن قيمة نصف القطر هى 50 . وإذا تم تغيير سطر 20 إلى

```
20 CIRCLE (160,100),50,,,,,2
```

فإن القطع الناقص يكون رأسياً بدلاً من كونه أفقياً .

مثال ١٢ - ٢٧ منطاد بنص متحرك Blimp with Animated Text

يحتوى شكل ١٢ - ٢٢ على برنامج ميكروسوفت بيسك متقدم مكتوب لحاسب IBM الشخصى لتوليد رسم منطاد . وعند مركز المنطاد توجد مساحة تعرض فيها رسالة متحركة (وفى هذه الحالة SCHAUM'S OUTLINE) .

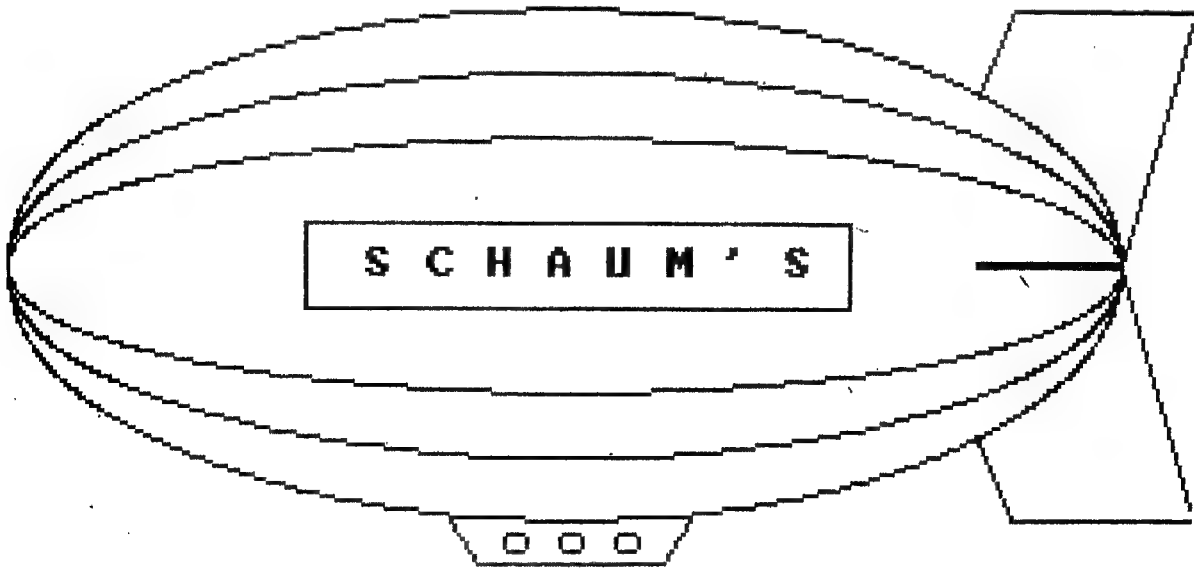
```
10 ***** BLIMP *****
20
30 KEY OFF : CLS
40 TEXT$="                S C H A U M ' S   O U T L I N E S   "
50 SCREEN 1 : COLOR 0,1
60
70 *** DRAW THE BLIMP ***
80
90 CIRCLE (150,100),150,3,,,,.4
100 CIRCLE (150,100),150,3,,,,.3
110 CIRCLE (150,100),150,3,,,,.2
120 LINE (260,60)-(270,40)
130 LINE -(320,40) : LINE -(300,100)
140 LINE (260,140)-(270,160)
150 LINE -(320,160) : LINE -(300,100)
160 LINE (260,99)-(300,99)
170 LINE (260,100)-(300,100)
180 LINE (118,160)-(125,170)
190 LINE -(175,170) : LINE -(182,160)
200 CIRCLE (135,165),3
210 CIRCLE (150,165),3
220 CIRCLE (165,165),3
230 LINE (80,90)-(225,110),3,B
240
250 *** GENERATE THE MESSAGE ***
260
270 FOR I=1 TO 52
280   LOCATE 13,12 : PRINT MID$(TEXT$,I,16)
290   FOR COUNT=1 TO 200 : NEXT COUNT
300   LOCATE 13,12 : PRINT SPACE$(16)
310 NEXT I
320 GOTO 270
330 END
```

شكل ١٢ - ٢٢

وتركيب هذا البرنامج هو في غاية البساطة . سطر 30 يسمح الشاشة و سطر 40 يخصص الرسالة للمتغير \$TEXT . ويحدد سطر 50 بيانات التحليل المتوسط بخلفية سوداء ولوحة ألوان رقم 1 (ألوان سماوى ، بنفسجى وأبيض) . والسطور من 90 إلى 230 تولد المنطاد نفسه . وعلى الأخص فإن السطور 90 ، 100 و 110 ، تولد ثلاثة قطاعات ناقصة مركزية وهى التى تكون الشكل العام للمنطاد . وتولد السطور 120 إلى 170 جزء الذيل . والسطور 180 إلى 230 تولد عجلات الهبوط تحت المنطاد وحدود مساحة العرض .

والسطور 270 إلى 320 تولد الرسالة المتحركة . لاحظ أن مجموعة الجمل هذه تتضمن استخدام دالة MID \$ داخل الحلقة التكرارية FOR TO . والفكرة الأساسية هى عرض 16 حرف من النص فى أى وقت واحد . مع كل ، فإن كل مرور خلال الحلقة التكرارية ينتج عنه نقل مجموعة الحروف بحرف واحد إلى اليمين . وكل المجموعة من الحروف يعقبها وقت تأخير قصير ثم مسح منطقة العرض استعداداً لمجموعة الحروف التالية . لاحظ أن سطر 230 يتسبب فى تكرار الحلقة التكرارية ويسبب ذلك استمرار الرسالة المتحركة بغير حدود .

ويبين شكل ١٢ - ٢٣ عرضاً نموذجياً ، وعلى ذلك فلا بد للقارئ أن يتذكر أن العرض فى حركة مستمرة عند تنفيذ البرنامج .



شكل ١٢ - ٢٣

وتحتوى أيضاً بعض نسخ بيسك الحاسب الدقيق على جملة DRAW التى تسمح بتعريف شكل معقد بدلالة سلسلة من الحروف . فمثلا جملة DRAW «R100 D50 L100 U50» تعرف مستطيل أفقى وذلك بالتحرك يمينا 100 وحدة و 50 وحدة إلى أسفل و 100 وحدة إلى اليسار ثم 50 وحدة إلى أعلى . ويولد تنفيذ هذه الجملة المستطيل أوتوماتيكياً

ولن نناقش جملة DRAW بأى تفاصيل فى هذا الكتاب حيث أن قواعد تعريف شكل السلاسل معقدة نوعاً ما . وعلى القارئ أن يرجع إلى كتيب بيسك المرجعى المتاح للحاسب الدقيق الخاص به لمزيد من المعلومات عن هذا الموضوع .

١٢ - ٤ الرسوم المتحركة ANIMATIONS

إن إحدى التطبيقات على الحاسبات الدقيقة الأكثر تشويقاً وتسلياً هى عروض الرسوم المتحركة . ومثل هذه التطبيقات تعطى أساساً للعديد من ألعاب الحاسبات وهى جزء مهم لكثير من البرامج التعليمية والفنية .

وقد قمنا فعلاً بعرض بعض الرسوم المتحركة البسيطة فى الأمثلة ١٢ - ٨ (نقط فى الفراغ) ، ١٢ - ١١ (الفن الحركى) ، ١٢ - ١٤ (المستطيلات المتحركة) ، ١٢ - ١٨ (الدوائر المتحركة) و ١٢ - ٢٧ (المنطاد بنص متحرك) . وكل هذه الرسوم المتحركة

تعتمد على نفس الفكرة : عرض أى شيء أو سطر في نص ، تكوين تأخير قصير للوقت ومسح الشيء . ثم التحرك إلى موضع قريب وإعادة السلسلة .

ويتم هذا القسم بالرسوم المتحركة التى تتضمن حركة أشياء مليئة . وسنرى أولاً كيف يتم ذلك باستخدام بعض الجمل المعروفة الآن مثل PAINT و CIRCLE وسنعرض طريقة بديلة تعتمد على جملتين جديدتين PUT,GET . وسنرى أن الطريقة الثانية تعطى بعض المزايا المتميزة عن الطريقة الأولى .

مثال ١٢ — ٢٨ محاكاة كرة مرتدة Simulation of a Bouncing Ball

دعنا الآن نعتبر رسم متحرك بسيط والذي يسمح لكرة بأن تتحرك بحرية داخل فراغ محدود . وينشأ الرسم المتحرك أولاً بعرض الكرة عند موضع معين وملئها بلون وتتوقف وقتاً قصيراً ثم تمسح الكرة . (أى يعاد عرض الكرة باستخدام لون الخلفية) . ثم تتحرك إلى موضع قريب وتكرر الخطوات كلها . وعندما يقابنا حاجز (أى حائط) فإن الكرة تغير اتجاهها وعلى ذلك تظهر الكرة وكأنها ترتد عن الحائط .

ويبين شكل ١٢ — ٢٤ برنامجاً كاملاً ميكروسوفت بيسك متقدماً مكتوباً لحاسب IBM الشخصى . وهذا البرنامج يستخدم بيانات التحليل العالى . وعلى ذلك فإنه لا يتضمن استخدام ألوان متعددة (يدعم حاسب IBM الشخصى أسود وأبيض فقط فى أسلوب التحليل العالى) . وفى هذا البرنامج يتم توليد موضع الكرة عند البداية عشوائياً وكذلك المسافة بين المواضع المتتالية .

```

10 ***** BOUNCING BALL *****
20
30 KEY OFF : CLS
40 DEFINT A-Z
50 RANDOMIZE : CLS
60 SCREEN 2
70 LINE (0,0)-(639,199),,B
80 LINE (10,5)-(629,194),,B
90 PAINT (5,2)
100 X=20+INT(600*RND) : Y=20+INT(160*RND)
110 DX=5*(INT(4*RND)+1) : DY=5*(INT(4*RND)+1)
120
130 *** BEGIN LOOP ***
140
150 CIRCLE (X,Y),10,0
160 PAINT (X,Y),0
170 X1=X+DX : Y1=Y+DY
180 IF X1 < 21 THEN X1=21 : DX=-DX : GOTO 200
190 IF X1 > 618 THEN X1=618 : DX=-DX
200 IF Y1 < 10 THEN Y1=10 : DY=-DY : GOTO 220
210 IF Y1 > 189 THEN Y1=189 : DY=-DY
220 CIRCLE (X1,Y1),10
230 PAINT (X1,Y1)
240 X=X1 : Y=Y1
250 GOTO 150
260 END

```

شكل ١٢ — ٢٤

دعنا نعتبر هذا البرنامج بالتفصيل . يحذف سطر 30 أى عروض سابقة لتعريفات مفتاح الدالة ويمسح الشاشة . و سطر 40 يقوم بتعريف كل المتغيرات لتكون من نوع الرقم الصحيح . و سطر 50 يستهل مولد العدد العشوائى ثم يمسخ الشاشة ، و سطر 60 يحدد بيانات التحليل العالى . تولد السطور 70 إلى 90 مستطيلاً مصمماً حول الحواف الخارجية للشاشة ، معطية بذلك العائق الذى يحتوى على حركة الكرة . ثم يتم توليد نقطة بداية عشوائية فى سطر 100 ، ويتم توليد مجموعة من القيم عشوائياً لزيادة كل حركة فى سطر 110 .

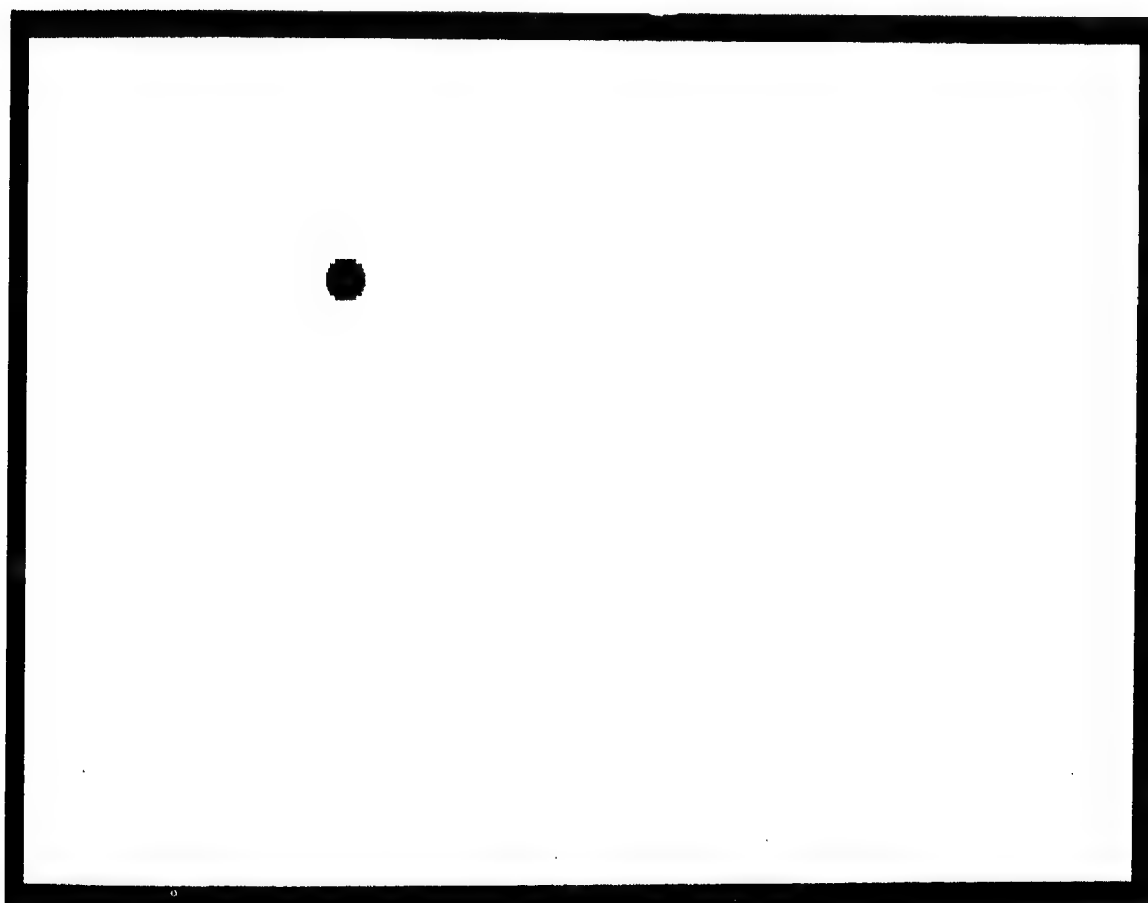
ويحتوى الجزء المتكرر من البرنامج على السطور 150 إلى 250 . ويسبب السطران 150 و 160 مسح الكرة من موضعها الحالى . (أى تعرض الكرة بلون أسود وهو لون الخلفية فى الموضع الحالى) . ثم يولد سطر 170 موضعاً جديداً . وفى السطور 180 إلى 210 يتم اختيار النقطة الجديدة

لتحديد ما إذا كانت الكرة ستذهب إلى ما بعد الحوايط . وإذا كان ذلك ، فإن اتجاه الحركة سيتغير وينشأ ذلك خداع بأن الكرة ارتدت عن الحوايط وتعود إلى الفراغ الداخلي المحدود .

يتسبب السطران 220 و 230 في عرض الكرة في الموضع الجديد ثم ملئها بلون أبيض ثم يتم تخفيض قيم الإحداثيات الجديدة للمتغير X و Y في سطر 240 . ويستخدم هذان المتغيران لمسح الكرة أثناء المرور التالى خلال الحلقة التكرارية .

وبين شكل ١٢ — ٢٥ نوع المخرجات التى تنشأ عند تنفيذ البرنامج . ومع كل ، فلا بد للقارئ أن يتذكر أن الكرة تظهر وكأنها فى حركة ثابتة عن تنفيذ البرنامج الحقيقى .

ورغم أن طريقة الرسوم المتحركة السابقة تعمل فى حالة جيدة ، فإن الرسوم المتحركة تكون بطيئة نسبياً ويصاحبها أحياناً ارتعاش مضايق . ومن الممكن لبعض نسخ ميكروسوفت بيسك أن تولد رسوماً متحركة أسرع وخالية من الارتعاش عن طريق استخدام جملتى GET و PUT . وبهذه الطريقة يتم توليد رسوم متحركة للشئ مرة واحدة . وتقوم جملة GET بتحويل الشئ فى الشاشة إلى صف ثم تقوم جملة PUT بعد ذلك بالتحويل من الكتلة المتراصة ثانياً إلى الشاشة عند موضع مختار . وإذا تم وضع الشئ مباشرة على نفسه ، فإنه يظهر بلون عكس اللون السابق (أى أسود يحل محل الأبيض) ، وعلى ذلك بتنفيذ زوج من PUT عند نفس الموضع فإنه يمكن عرض الشئ أوتوماتيكياً ثم مسحه (لاحظ أنه عند استخدام جملتى GET و PUT بهذه الطريقة لا يحدث خلط بجملتى GET و PUT المستخدمة فيما يتعلق بملفات البيانات العشوائية كما هو مبين فى قسم ٩ — ٤ ، مثال (٩ — ٣١) .



والطريقة العامة إذاً هي توليد الشيء الذى يتحرك حول الشاشة ووضعه فى كتلة متراسة باستخدام جملة GET . وهذا يتم مرة واحدة فقط ويدخل البرنامج فى حلقة تكرارية والتي يتم فيها مسح الشيء من مكانه القديم (عن طريق جملة PUT) ، ويحدد موضع جديد ويعاد عرض الشيء فى هذا الموضع الجديد (ثانية عن طريق PUT) والرسم المتحرك ينشأ عن طريق تكرار المرور خلال الحركة التكرارية .

وتتكون جملة GET من كلمة GET يعقبها زوجان من الإحداثيات . ولابد أن ينحصر كل زوج من الإحداثيات بين قوسين تفصلهما فاصلة . وتقوم هذه الإحداثيات بتعريف الأركان المتقابلة للمستطيل الذى يحتوى على الشيء . ويفصل الشرطة (علامة ناقص) زوجى الإحداثيات . ويتبع الزوج الثانى من الإحداثيات اسم الكتلة المتراسة الذى يحوى الشيء ، أى GET (X,Y)-(X+20,Y+20), FIGURE . وتعتمد أبعاد الكتلة المتراسة على كل من حجم الشيء (حقيقة حجم المستطيل المحصور) ومستوى التحليل . وهناك صيغة رياضية لحساب الأبعاد المطلوبة التى تستخدم هذين العاملين . وعلى القارئ أن يرجع إلى الكتيب المرجعى لحاسبه الخاص لمعلومات عن هذا الموضوع . وبمجرد تخزين الشيء فى الكتلة المتراسة المطلوبة يتم إعادة عرضه بواسطة جملة PUT . وتتكون هذه الجملة من كلمة PUT يعقبها زوج مفرد من الإحداثيات محصورة بين قوسين وتفصلهما فاصلة . وتمثل هذه الإحداثيات الركن الأعلى الأيسر فى المثلث الذى يحتاج للشيء . ويعقب الإحداثيات بعد ذلك اسم الكتلة المتراسة أى PUT (X,Y),FIGURE

مثال ١٢ — ٢٩ العودة إلى الكرة المرتدة The Bouncing Ball Revisited

دعنا نعتبر ثانياً الرسم المتحرك لكرة داخل سياج كما هو موضح فى مثال ١٢ — ٢٨ . ومع كل سنقوم بتوليد الرسم المتحرك باستخدام جمل GET و PUT .

يحتوى شكل ١٢ — ٢٦ على البرنامج الكامل مكتوب مرة أخرى بميكروسوفت بيسك المتقدم لحاسب IBM الشخصى . ويشبه هذا البرنامج ذلك الموضح فى شكل ١٢ — ٢٤ . ومع كل ، فهناك بعض الاختلافات الهامة . لاحظ إضافة جملة DIM فى سطر 70 ، والذى يقوم بتعريف BALL على أنها كتلة متراسة من 34 عنصر . لاحظ أيضاً أن المسافة بين موضعين متتابعين (تم توليدها فى سطر 120) تتناقص حتى تستوى حركة الكرة . ولم يكن هذا عملياً فى النسخ السابقة ، لأن الرسم المتحرك كان بطيئاً إلى حد ما .

```

10 ***** BOUNCING BALL *****
20
30 KEY OFF : CLS
40 DEFINT A-Z
50 RANDOMIZE : CLS
60 SCREEN 2
70 DIM BALL(34)
80 LINE (0,0)-(639,199),,B
90 LINE (10,5)-(629,194),,B
100 PAINT (5,2)
110 X=20+INT(600*RND) : Y=20+INT(160*RND)
120 DX=1+INT(10*RND) : DY=1+INT(10*RND)
130 CIRCLE (X,Y),10
140 PAINT (X,Y)
150 GET (X-10,Y-10)-(X+10,Y+10),BALL
160
170 *** BEGIN LOOP ***
180
190 PUT (X-10,Y-10),BALL
200 X1=X+DX : Y1=Y+DY
210 IF X1 < 21 THEN X1=21 : DX=-DX : GOTO 230
220 IF X1 > 618 THEN X1=618 : DX=-DX
230 IF Y1 < 10 THEN Y1=10 : DY=-DY : GOTO 250
240 IF Y1 > 189 THEN Y1=189 : DY=-DY
250 PUT (X1-10,Y1-10),BALL
260 X=X1 : Y=Y1
270 FOR C=1 TO 5: NEXT C
280 GOTO 190
290 END

```

تظهر الجملتان CIRCLE و PAINT مرة واحدة فقط في البرنامج الحالي في سطرى 130 و 140 على الترتيب . تعطى هاتان الجملتان تعريف شكل الكرة في البداية . وتنسب جملة GET في سطر 150 بتخزين تعريف هذا الشكل في كتلة متراصة تسمى BALL .

ويختلف جزء الحلقة التكرارية في هذا البرنامج أيضاً بعض الشيء عن النسخة السابقة . ويتم استبدال أزواج جملتى CIRCLE و PAINT بجمل PUT في سطرى 190 و 250 . ويمسح الأول منهما في سطر 190 الكرة في الموضع السابق . ويقوم الثانى بعرض الكرة في موضعها الجديد . لاحظ في النهاية أن الحلقة التكرارية الفارغة FOR-TO قد تمت إضافتها في سطر 270 . ويهدف هذه الحلقة التكرارية إلى إبطاء الرسم المتحرك حتى يكون أكثر تشويقاً للعين .

وسوف تظهر الشاشة مرة أخرى عند تنفيذ البرنامج في شكل ١٢ — ٢٥ إلا أن الحركة الآن ستكون أكثر سرعة وأكثر تسوية . ونسب بالقرارى أن يقوم بالتنفيذ الفعلى لكل من البرنامجين حتى يكون أكثر تفهماً للاختلافات في الرسوم المتحركة الناتجة .

ولا تكون الرسوم المتحركة التى تتكون بهذه الطريقة محدودة بشئ واحد متحرك إذ يمكن أيضاً تحريك عدة أشياء حول الشاشة باستخدام أزواج متعددة من جمل PUT . ولابد من تخزين كل شئ في كتلة متراصة منفصلة . ويبين المثال التالى هذه الطريقة بالتفصيل .

مثال ١٢ — ٣٠ لعبة كرة التجديف A Game of Paddleball

هذه نسخة بسيطة من لعبة فيديو شائعة يشار إليها عادة بكرة التجديف . تنحصر الكرة بين ثلاث حوائط . وهناك مجذاف متحرك صغير موضوع حيث كان يجب وضع الحائط الرابع عادة . ويمكن تحريك هذا الجذاف إلى أعلى أو أسفل استجابة إلى وضع أحد وسائل التحكم مثل عصا التوجيه (سوف نفترض أن الحائط الغير موجود هو رأسى) . وإذا ما اصطدمت الكرة بالجذاف ، فإنها تعود ثانية إلى منطقة اللعب ، وإلا فإن الكرة تمر خلال الفتحة وتختفى .

والغرض من اللعبة هو التعرف مسبقاً على مسار الكرة حسباً تتحرك تجاه المنطقة المفتوحة ووضع الجذاف بحيث يصدم الكرة ويعيدها ثانية إلى منطقة اللعب . وسيحصل اللاعب على نقطة لكل صدمة للكرة ويحسر نقطة واحدة إذا أخطأت الكرة الجذاف . وسوف تعود الكرة في الظهور تلقائياً عند موضع ما عشوائياً داخل منطقة اللعب بعد كل مرة تخطئ فيها الاصطدام مع الجذاف .

يحتوى شكل ١٢ — ٢٧ على برنامج يبسك كامل لهذه اللعبة . والبرنامج مكتوب بميكروسوفت بيسك المتقدم لحاسب IBM الشخصى . ويحتوى على استخدام بيانات التحليل المتوسط ، جمل تحكم عصا التوجيه وتحسينات الصوت لمصاحبة حركة الكرة . وعلى ذلك فإن البرنامج يحتوى على العديد من الخصائص الموضحة في فصل ١٠ وكذلك طريقة الرسوم المتحركة .

دعنا الآن نعتبر الخصائص الرئيسية لهذا البرنامج . يحدف سطر 30 أى تعريفات سابقة لمفتاح الدالة ، ويمسح الشاشة . ويعلم سطر 40 عن كل المتغيرات من نوع الرقم الصحيح . وتولد السطور 50 إلى 70 عرض نص مبدئى . ويستهل سطر 80 مولد العدد العشوائى . وسطر 90 يجعل الإشارة القادمة من زر 1 لعصا التوجيه مصاحباً للبرنامج الفرعى ، ثم يقوم بتشغيل هذه المصاحبة (يستخدم زر 1 لإيقاف هذا البرنامج) . ويستدعى سطر 100 بيانات التحليل المتوسط مع خلفية سوداء ولوحة ألوان رقم 1 (ألوان سماوى ، بنفسجى وأبيض) . ويتم إدخال الكتل المتراصة التى تحتوى على تعريفات شكل الكرة والجذاف في سطر 110 ، ويتم تخصيص القيم الابتدائية لبعض معالم البرنامج في سطر 120 (لاحظ أن DX و DY تحددان السرعة الرأسية والأفقية للكرة كما هو موضح في المثالين السابقين . وتحدد P موضع الجذاف) .

وتولد السطور 160 إلى 180 الجدران الثلاثة ، وينسب السطران 190 و 200 عرضاً مختصراً للنص عند أسفل الشاشة تحت منطقة العرض البياضى . ويحتوى النص على النتيجة وعلى تلقين يبين كيف تنتهى اللعبة .

وسطر 240 يتسبب في توليد موضع الكرة عند البداية عشوائياً . ثم يتم توليد شكل الكرة وشكل الجذاف ، ثم يتم تخزينهما في الكتلتين المتراصتين المناظرتين في السطرين 250, 260 .

وتحتوى الحلقة التكرارية الرئيسية التى ينشأ فيها الرسم المتحرك على السطور 300 إلى 440 ويحدد سطر 300 المواضع الجديدة للكرة والجذاف . ويضبط السطران 310 و 320 موضع الجذاف إذا ما كان أعلى ارتفاعاً أو أقل انخفاضاً من اللازم . ويختبر السطران 330 و 340 موضع الكرة لمعرفة ما إذا كان يقع داخل مسار الجذاف . وإذا كان كذلك فإنه يتم تداول برنامجين فرعيين مختلفين : ويتوقف الاختيار عما إذا كان الجذاف يصدم الكرة أم لا حسب ما يحدده الاختبار في سطر 340 .

```

10 ***** PADDLEBALL GAME *****
20
30 KEY OFF : CLS
40 DEFINT A-Z
50 LOCATE 4,1 : PRINT "Welcome to PADDLEBALL"
60 LOCATE 8,1 : PRINT "Rules: 1 point for each hit, "
70 LOCATE 10,7 : PRINT "-1 point for each miss"
80 LOCATE 14,1 : RANDOMIZE : CLS
90 ON STRIG(0) GOSUB 610 : STRIG(0) ON
100 SCREEN 1 : COLOR 0,1
110 DIM BALL(34),PADDLE(22)
120 DX=10 : DY=10 : P=8 : SCORE=0
130
140 ***** DRAW THE BORDER *****
150
160 LINE (0,0)-(319,5),1,BF
170 LINE (0,178)-(319,183),1,BF
180 LINE (314,6)-(319,177),1,BF
190 LOCATE 25,1 : PRINT "Score:";SCORE;
200 LOCATE 25,20 : PRINT "To stop, press B1";
210
220 ***** DRAW THE INITIAL FIGURES *****
230
240 X=10+INT(300*RND) : Y=10+INT(164*RND)
250 CIRCLE (X,Y),5 : PAINT (X,Y) : GET (X-5,Y-5)-(X+5,Y+5),BALL
260 LINE(0,P)-(6,P+20),3,BF : GET (0,P)-(6,P+20),PADDLE
270
280 ***** MAIN LOOP *****
290
300 X1=X+DX : Y1=Y+DY : DUMMY=STICK(0) : P1=199*STICK(1)/100
310 IF P1 < 6 THEN P1=6 : GOTO 330
320 IF P1 > 157 THEN P1=157
330 IF X1 >= 12 THEN 350
340 IF Y1 >= P1-2 AND Y1 <= P1+22 THEN GOSUB 480 ELSE GOSUB 530
350 IF X1 > 308 THEN X1=308 : DX=-DX : SOUND 1000,2
360 IF Y1 < 11 THEN Y1=11 : DY=-DY : SOUND 1000,2 : GOTO 380
370 IF Y1 > 172 THEN Y1=172 : DY=-DY : SOUND 1000,2
380 PUT (X-5,Y-5),BALL : 'erase ball
390 PUT (0,P),PADDLE : 'erase paddle
400 PUT (X1-5,Y1-5),BALL : 'redraw ball
410 PUT (0,P1),PADDLE : 'redraw paddle
420 X=X1 : Y=Y1 : P=P1
430 FOR C=1 TO 10: NEXT C
440 GOTO 300
450
460 ***** PADDLE HIT THE BALL *****
470
480 X1=12 : DX=-DX : SCORE=SCORE+1 : SOUND 1000,2
490 LOCATE 25,7 : PRINT SCORE; : RETURN 360
500
510 ***** PADDLE MISSED THE BALL *****
520
530 IF X1 >= 5 THEN RETURN 360
540 PUT (X-5,Y-5),BALL : PUT (0,P),PADDLE
550 DX=10 : DY=10 : SCORE=SCORE-1 : SOUND 50,2
560 LOCATE 25,7 : PRINT SCORE;
570 FOR I=1 TO 1000 : NEXT I : RETURN 240
580
590 ***** END THE GAME *****
600
610 END

```

وتختبر السطور 350 إلى 370 موضع الكرة بالنسبة إلى الجدران الثابتة الثلاثة . فإذا ما تداخل موضع الكرة المحسوب مع موضع الجدار فإنه يتم إجراء تعديل في موضع الكرة كما يتم انعكاس سرعتها مكوناً بذلك خداعاً بأن الكرة ارتدت عن الحائط . ويصاحب الارتداد صوتاً ذا نغمة عالية .

ويسبب السطران 380 و 390 مسح الأشياء من موضعها القديم . ويسبب السطران 400 و 410 إعادة رسمها في موضعها الجديد . ويستهل سطر 420 قيم المواضع القديمة ، ويدخل في سطر 430 تأخير وقت قصير وذلك تجنباً للرعشات الزائدة أساساً . ويعيد سطر 440 التحكم إلى سطر 300 ، وهكذا يبدأ مرور آخر خلال الحركة التكرارية .

ويقوم السطران 480 و 490 بعمل برنامج فرعى . يتم تداوله عندما يضرب الجدار الكرة . وينشئ هذا البرنامج الفرعى أساساً تأثير ارتداد مشابه لما ينتج عندما تلمس الكرة أحد الجدران . وتزداد الأهداف المتراكمة مقدار 1 ، ويتم عرض القيم النهائية عند أسفل الشاشة .

وتقوم السطور 530 إلى 570 بعمل برنامج فرعى يمكن تداوله عندما يخطئ الجدار الكرة . ويعرض السطر 530 الكرة ببساطة بالقرب من الحافة اليسرى للشاشة إذا ما كان هذا هو موضعها المحسوب ، وبهذا يمنع الكرة من الاختفاء عن النظر قبل الآوان .

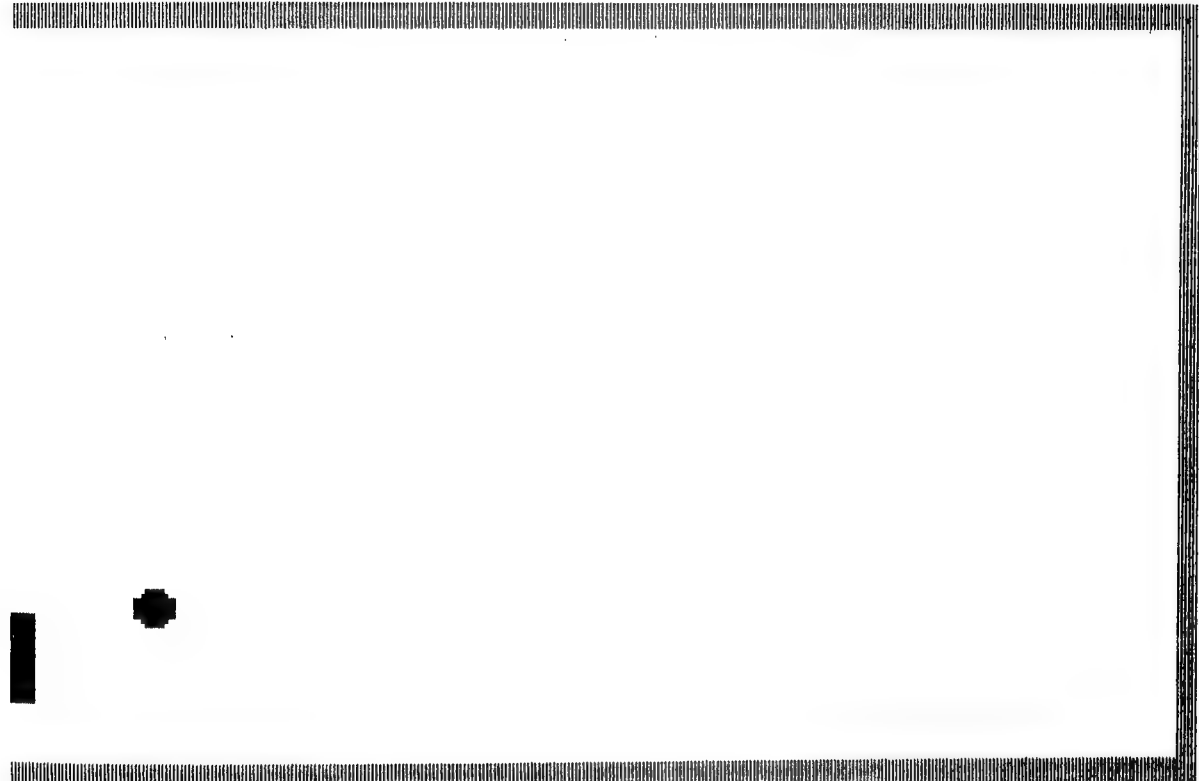
Welcome to PADDLEBALL

Rules: 1 point for each hit,

-1 point for each miss

Random number seed (-32768 to 32767)? 12345

شكل ١٢ - ٢٨



Score: -5

To stop, press B1

شكل ١٢ - ٢٩

ويقوم بمحاكاة تأثير اختفاء الكرة عن النظر في سطر 540. ويقوم نفس السطر بمسح كل من الكرة والمجداف. ويسبب سطر 550 إعادة وضع السرعات إلى قيمتها الأولية الموجبة مسبباً بذلك وصول الكرة إلى أسفل الركن الأيمن للشاشة عندما تعود للظهور عند موضع جديد عشوائي.

وتتناقص الأهداف المتجمعة بمقدار 1 ويتم توليد صوت قصير ذي نغمة منخفضة (لاحظ أنه يصدر صوت ذو نغمة مرتفعة عندما ترتد الكرة عن سطح ولكننا نسمع صوتاً ذا نغمة منخفضة عندما يخطئ المجداف الكرة). ثم يتم عرض الهدف الجديد عند أسفل الشاشة (سطر 560)، ويتم توليد تأخير وقت قصير (سطر 570) قبل عودة التحكم إلى سطر 240 مسبباً وضع كرة جديدة في اللعب.

وفي النهاية فإن جملة END في سطر 610 هي في الحقيقة البرنامج الفرعي الذي يتم تداوله عندما يتم الضغط على زر 1 على عصا التوجيه. وهذا التصاحب هو الذي تم تنشيطه في سطر 90.

وشكل ١٢ — ٢٨ يبين عرض النص عند أول ظهور له عند تنفيذ البرنامج. وبمجرد وجود قيمة لمولد العدد العشوائي فإن الشاشة تتحول إلى بيانات التحليل المتوسط مبنية جدران سماوية وكرة بيضاء ومجداف. وتظهر الكرة عند وضع عشوائي على الشاشة وتظل في حركة ثابتة وترتد عن أى سطح تلمسه. ويبين شكل ١٢ — ٢٩ مظهر الشاشة كلما اقتربت الكرة من المجداف في أعلى اليمين.

ومن خلال كل هذا فإن اللاعب يتحكم في موضع المجداف وذلك بضبط عصا التوجيه. ويستمر تكرار حركة الارتداد حتى يتم السماح للكرة بالاختفاء في الجانب الأيسر للشاشة. ويرجع ذلك لتحديد موضع غير صحيح للمجداف، ثم تعود الكرة للظهور عند موضع جديد بعد فترة قصيرة.

وتترايد الأهداف المتجمعة بمقدار 1 في كل مرة يضرب المجداف الكرة، وبالمثل فإن الأهداف المتجمعة تتناقص بمقدار 1 كلما أخطأ المجداف الكرة. ويتم دائماً عرض القيمة الحالية للأهداف المتجمعة في أسفل الشاشة، وتستمر اللعبة حتى يضبط اللاعب على زر 1 لعصا التوجيه.

ونهب بالقرىء أن يقوم بتنفيذ هذا البرنامج إذا أمكن حتى يمكن تفهم ماذا يحدث حقيقة. وهذا طبعاً مرغوب فيه جداً لكل البرامج من هذا النوع.

١٢ — ٥ بيانات الحروف CHARACTER GRAPHICS

يمكن الحصول على تأثيرات بيانية معينة غالباً عن طريق معالجة ذكية لحروف النص. وتحتوي مثل هذه التأثيرات على عرض رسائل متحركة، وتكوين أشكال ورسوم بسيطة. وقد رأينا فعلاً بعض الأمثلة من عروض الرسائل المتحركة بلف رأسى في المائتين ١٠ — ١٠٠ و ١٧ — ١٢ وتحرك أفقى كجزء من مثال ١٢ — ٢٧.

ويبين برنامج ارتداد الكرة في مثال ٦ — ٢٨ طريقة بسيطة ولكنها فعالة لاستخدام بيانات الحروف على طابع السطور. ويسمح البرنامج الموضح في مثال ١٠ — ١٣ باستخدام عصا توجيه لتكوين أشكال مختلفة كتجمعات من المنجمة (النجوم). وكقاعدة فإن التأثيرات البيانية التي تنشأ بهذه الطريقة تكون أقل فعالية عن تلك التي تنشأ بوسيلة البيانات. ومع كل فإن مثل تأثير البيانات هذه يخدم غرضاً خصوصاً للحاسبات الدقيقة التي لا تدعم النص المنفصل ووسائل البيانات.

ويوضح المثال التالى استخدام نوع بيانات الحروف لتمثيل البيانات الرقمية.

مثال ١٢ — ٣١ مولد خريطة الأعمدة A Barchart Generator

نفرض أن لدينا مجموعة من القيم العددية التي نريد عرضها في شكل بياني لتأكيد الطبيعة المنفصلة لكل قيمة. وغالباً تكون خريطة الأعمدة البيانية هي أفضل طريقة يتم بها ذلك والتي تمثل فيها كل قيمة بمستطيل يكون ارتفاعه متناسباً مباشرة مع قيمته المناظرة وعادة ما تعرض المستطيلات أفقياً وتكون كلها مرئية في نفس الوقت.

وشكل ١٢ — ٣٠ يحتوى على برنامج ميكروسوفت بيسك مكتوب لحاسب IBM الشخصى والذي يتسبب في توليد خريطة أعمدة بيانية لعدد يصل إلى ١٢ قيمة عددية. ومن المفروض أن تكون كل قيمة غير سالبة (أى أكبر من أو مساوية للصفر). وسيتم عرض القيم أعلى كل مستطيل مناظر لها (أى أعلى العمود) وستكون لكل قيمة عنوان مناظر يتم عرضه تحت العمود. وستكون الأعمدة نفسها مكونة من تجمعات من النجوم.

```

10 '***** BAR CHART GENERATOR *****
20 '
30 KEY OFF: CLS
40 WIDTH 80: SCREEN 0: COLOR 7,0
50 DIM LABEL$(12),Y(12)
60 LOCATE 1,20: PRINT STRING$(40,"*")
70 LOCATE 2,20: PRINT "*";SPC(38);"*"
80 LOCATE 3,20: PRINT "*";SPC(9);"BAR CHART GENERATOR";SPC(10);"*"
90 LOCATE 4,20: PRINT "*";SPC(38);"*"
100 LOCATE 5,20: PRINT STRING$(40,"*")
110 YMAX=0
120 '
130 '***** ENTER DATA AND FIND LARGEST Y *****
140 '
150 LOCATE 7,1: INPUT "Title: ",TITLE$
160 LOCATE 9,1: INPUT "How many data items? (1-12) ",ANS$: N=VAL(ANS$)
170 IF N < 1 OR N > 12 THEN BEEP: LOCATE 9,29: PRINT SPACE$(6): GOTO 160
180 FOR I=1 TO N
190     LOCATE I+10: PRINT "I =",I
200     LOCATE I+10,15: INPUT "Value: ",ANS$: Y(I)=VAL(ANS$)
210     IF LEFT$(ANS$,1)="0" THEN 230
220     IF Y(I) <= 0 THEN BEEP: LOCATE I+10,21: PRINT SPACE$(6): GOTO 200
230     LOCATE I+10,35: INPUT "Label: ",LABEL$(I)
240     IF Y(I) > YMAX THEN YMAX=Y(I)
250 NEXT I
260 '
270 '***** GENERATE AND DISPLAY THE BAR CHART *****
280 '
290 CLS: W=60\N
300 FOR I=1 TO N
310     R=20-18*Y(I)\YMAX: IF R=20 THEN 370
320     FOR ROW=R TO 20
330         FOR COL=(I-1)*W+11 TO I*W+8
340             LOCATE ROW,COL: PRINT "*"
350         NEXT COL
360     NEXT ROW
370 NEXT I
380 '
390 '***** LABEL THE BAR CHART *****
400 '
410 FOR I=1 TO N
420     R=19-18*Y(I)\YMAX
430     C=9 + (I-1)*W + (W-LEN(STR$(Y(I))))/2
440     LOCATE R,C: PRINT Y(I)
450     C=10 + (I-1)*W + (W-LEN(LABEL$(I)))/2
460     LOCATE 21,C: PRINT LABEL$(I)
470 NEXT I
480 C=10 + (N*W-LEN(TITLE$))/2
490 LOCATE 23,C: PRINT TITLE$: LOCATE 23,1
500 END

```

Annual Sales Increases

Increase (%)	Year
5.2	1985
7.8	1986
8.2	1987
6.7	1988
10.6	1989
12.3	1990

دعنا نختبر البرنامج ببعض التفصيل . تسمح السطور 30 إلى 110 الشاشة . وتقوم بتعريف الكتل المتراصة LABEL\$,Y وتوليد عنوان واستهلال YMAX . وتولد السطور 150 إلى 250 سلسلة من التلقينات التفاعلية للبيانات الداخلة .

وأول بند مطلوب هو عنوان خريطة الأعمدة البيانية (سطر 150) يعقبه طلب إدخال عدد بنود البيانات (السطران 160 و 170) . لاحظ أن هذا الطلب يحتوي على تصيد خطأ (سطر 170) والذي يحتاج إلى استجابة المستخدم لتكون قيمته بين 1 و 12 . وتقوم السطور 180 إلى 250 بعمل حلقة تكرارية FOR-TO والتي تلقن المستخدم لكل من بنود البيانات وكل عنوان يصاحبها . وتحتوي الحلقة التكرارية على تصيد خطأ (السطران 210 و 220) الذي يمنع بعض الحروف بخلاف الأعداد من إدخالها للقيمة المطلوبة . وأيضاً فإن أكبر قيمة مدخلة توجد في سطر 210

و يتم توليد الأعمدة في السطور 290 إلى 370 ويسمح سطر 290 للبيانات المدخلة ويحسب عرض كل عمود . (وكلما زاد عدد الأعمدة صغر عرض كل عمود) . وتقوم السطور 300 إلى 370 بعمل حلقة تكرارية ثلاثية تستخدم لتوليد الأعمدة وتنسب الحلقة الخارجية في توليد الأعمدة المتتالية (لاحظ أن النصف الأعلى لكل عمود والمحسوب في سطر 310 يحسب على أساس النسبة بين القيمة المعطاة والقيمة العظمى . وتولد الحلقة التكرارية الوسطى (السطور 320 إلى 360) الصفوف التي تكون كل عمود وتولد الحلقة التكرارية الداخلة (السطور 330 إلى 350) الأعمدة داخل كل صف .

وتقوم السطور 410 إلى 490 بعنوان خريطة الأعمدة . وتسبب الحلقة التكرارية FOR-TO (السطور 410 إلى 470) في عرض قيم كل بيان على السطر أعلى كل عمود وظهور العنوان المناظر تحت كل عمود . ويتسبب السطران 480 و 490 في عرض عنوان الخريطة في الأعمدة الأسفل أفقياً وعند المركز .

ونفرض الآن برنامجاً يتم تنفيذه باستخدام مجموعة البيانات المدخلة التالية : وبين شكل ١٢ - ٣١ الديالوج الذي تم توليده بجزء البرنامج للبيانات المدخلة (واستجابة المستخدم تحتها خط) . وشكل ١٢ - ٣٢ بين خريطة الأعمدة البيانية المناظرة .

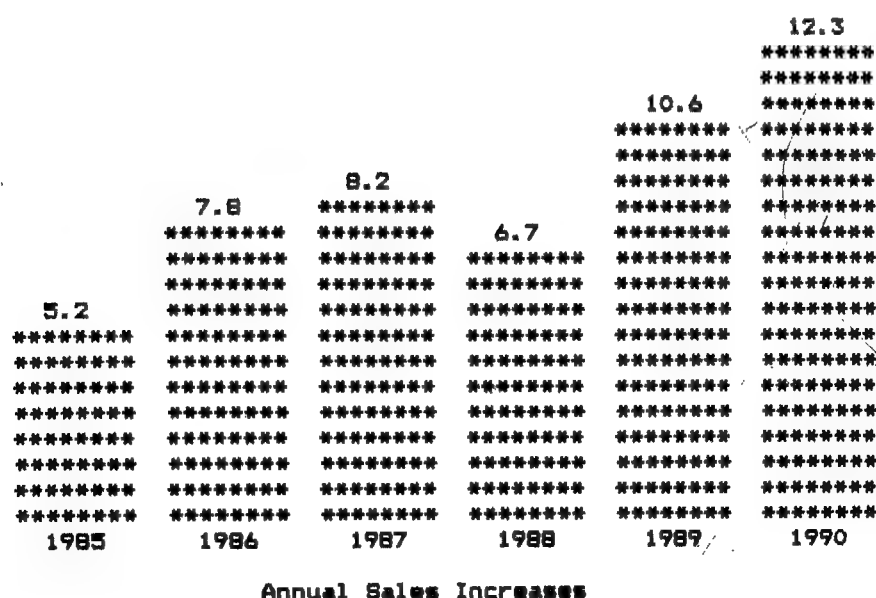
```
*****
*                                     *
*               BAR CHART GENERATOR               *
*                                     *
*****
```

Title: Annual Sales Increases

How many data items? (1-12) 6

I = 1	Value: <u>5.2</u>	Label: <u>1985</u>
I = 2	Value: <u>7.8</u>	Label: <u>1986</u>
I = 3	Value: <u>8.2</u>	Label: <u>1987</u>
I = 4	Value: <u>6.7</u>	Label: <u>1988</u>
I = 5	Value: <u>10.6</u>	Label: <u>1989</u>
I = 6	Value: <u>12.3</u>	Label: <u>1990</u>

وتحتوي بعض الحاسبات الدقيقة على مجموعة خاصة من بيانات الحروف بالإضافة إلى 128 حرف ASCII قياسياً المبينة في (ملحقه). فمثلاً يدعم حاسب IBM الشخصي 256 حرف مختلف متضمناً 128 حرفاً خاصاً. والعديد من هذه هي حروف يابانية (رموز يابانية والحروف المستخدمة في اللغات الأجنبية). وقد استخدمنا فعلاً أحد بيانات الحروف هذه لتكوين كتل من الضوء في مثال ١٢ - ١٠ (انظر سطر 50 في شكل ١٠ - ٧). واستخدام بيانات الحروف هذه يعطي إمكانية تحسين معنى لعروض يابانية.



شکل ۱۲ - ۳۲

مثال ١٢ - ٣٢ مولد محسن بخريطة الأعمدة البيانية ، An Improved Barchart Generator

يحتوي شكل ١٢ - ٣٢ على برنامج آخر لميكروسوفت بيسك مكتوب لحاسب IBM الشخصي لتوليد خريطة أعمدة بيانية كما هو مبين في مثال ١٢ - ٣١ . هذا البرنامج يشبه البرنامج المعروض في شكل ١٢ - ٣٠ . ونرى الآن إشارة إلى بيانات الحروف الخاصة متاحة على حاسب IBM الشخصي . CHR\$ (200) و CHR\$ (186) و CHR\$ (187) و CHR\$ (205) و CHR\$ (201) .

لاحظ مثلاً أن السطور 60 إلى 100 تحتوي على دالة تستدعي ضمن جمل PRINT المختلفة . واستدعاء هذه الدالة يتسبب في وضع العنوان «BAR CHART GENERATOR» داخل برواز محاط بمستطيل مزدوج كما هو مبين في شكل ١٢ - ٣٤ .

ويحدد سطر 110 أيضاً الحروف (177) CHR\$ ، (178) CHR\$ للمتغير \$BARI و\$BAR2 على الترتيب. ويتم طبع الحروف في سطر 350 مسببة بذلك تكوين الأعمدة المستطيلة من كل مستطيلة مظلة جزئياً بدلاً من النجوم . ويبين شكل ١٢ - ٣٥ هذا التأثير . لاحظ أن الأعمدة المتناوبة مبنية بتظليل مختلف .

ومن الأمور الشيقة مقارنة شكل ١٢ - ٣١ مع شكل ١٢ - ٣٤ . وواضح أن بيانات الحروف الخاصة المستخدمة لتوليد شكل ١٢ - ٤٠ يحسن مظهر العرض . والأكثر إلفاً للنظر هو مقارنة شكل ١٢ - ٣٢ مع ١٢ - ٣٥ . (شكل ١٢ - ٣٥ يظهر أفضل على شاشة TV حيث كتلة الحروف المفردة داخل كل عمود سوف تلمس احدها الآخر مكونة مستطيلاً مصمماً) وعلى ذلك فإننا نرى أنه يمكن الحصول على عرض أفضل كثيراً في الرؤية بإضافة جهد إضافي صغير من الريجة .

```

10 '***** BAR CHART GENERATOR *****
20 '
30 KEY OFF: CLS
40 WIDTH 80: SCREEN 0: COLOR 7,0
50 DIM LABEL$(12),Y(12)
60 LOCATE 1,20: PRINT CHR$(201);STRING$(38,CHR$(205));CHR$(187)
70 LOCATE 2,20: PRINT CHR$(186);SPC(38);CHR$(186)
80 LOCATE 3,20: PRINT CHR$(186);SPC(9);"BAR CHART GENERATOR";SPC(10);CHR$(186)
90 LOCATE 4,20: PRINT CHR$(186);SPC(38);CHR$(186)
100 LOCATE 5,20: PRINT CHR$(200);STRING$(38,CHR$(205));CHR$(188)
110 BAR1$=CHR$(177): BAR2$=CHR$(178): YMAX=0
120 '
130 '***** ENTER DATA AND FIND LARGEST Y *****
140 '
150 LOCATE 7,1: INPUT "Title: ",TITLE$
160 LOCATE 9,1: INPUT "How many data items? (1-12) ",ANS$: N=VAL(ANS$)
170 IF N < 1 OR N > 12 THEN BEEP: LOCATE 9,32: PRINT SPACE$(6): GOTO 160
180 FOR I=1 TO N
190     LOCATE I+10: PRINT "I =";I
200     LOCATE I+10,15: INPUT "Value: ",ANS$: Y(I)=VAL(ANS$)
210     IF LEFT$(ANS$,1)="0" THEN 230
220     IF Y(I) <= 0 THEN BEEP: LOCATE I+10,21: PRINT SPACE$(6): GOTO 200
230     LOCATE I+10,35: INPUT "Label: ",LABEL$(I)
240     IF Y(I) > YMAX THEN YMAX=Y(I)
250 NEXT I
260 '
270 '***** GENERATE AND DISPLAY THE BAR CHART *****
280 '
290 CLS: W=60\N
300 FOR I=1 TO N
310     R=20-18*Y(I)\YMAX: IF R=20 THEN 380
320     FOR ROW=R TO 20
330         FOR COL=(I-1)*W+11 TO I*W+8
340             LOCATE ROW,COL
350             IF I MOD 2 = 0 THEN PRINT BAR1$ ELSE PRINT BAR2$
360         NEXT COL
370     NEXT ROW
380 NEXT I
390 '
400 '***** LABEL THE BAR CHART *****
410 '
420 FOR I=1 TO N
430     R=19-18*Y(I)\YMAX
440     C=9 + (I-1)*W + (W-LEN(STR$(Y(I))))/2
450     LOCATE R,C: PRINT Y(I)
460     C=10 + (I-1)*W + (W-LEN(LABEL$(I)))/2
470     LOCATE 21,C: PRINT LABEL$(I)
480 NEXT I
490 C=10 + (N*W-LEN(TITLE$))/2
500 LOCATE 23,C: PRINT TITLE$: LOCATE 23,1
510 END

```

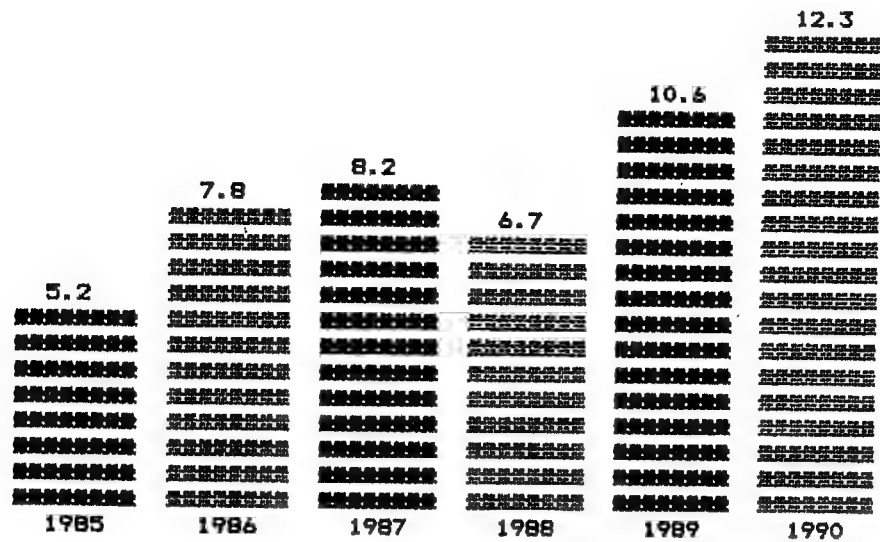
BAR CHART GENERATOR

Title: Annual Sales Increases

How many data items? (1-12) 6

I = 1	Value: <u>5.2</u>	Label: <u>1985</u>
I = 2	Value: <u>7.8</u>	Label: <u>1986</u>
I = 3	Value: <u>8.2</u>	Label: <u>1987</u>
I = 4	Value: <u>6.7</u>	Label: <u>1988</u>
I = 5	Value: <u>10.6</u>	Label: <u>1989</u>
I = 6	Value: <u>12.3</u>	Label: <u>1990</u>

شکل ۱۲ - ۳۴



Annual Sales Increases

شکل ۱۲ - ۳۵

اسئلة للمراجعة

Review Questions

- ١ - ١٢ حدد أن كانت أولا نسخة البيسك المتاحة لحاسبك الدقيق الخاص تحوى جمل بيانات خاصة . واذا كانت كذلك هل هى نفسها كما تم شرحها فى هذا الفصل ؟
- ٢ - ١٢ ما هو مستوى التحليل المعد لخواص البيانات فى حاسبك الدقيق الخاص ؟ هل هناك انماط بيانات عديدة متاحة ؟
- ٣ - ١٢ هل اللون متاح على حاسبك الدقيق الخاص ؟ اذا كان كذلك فكى عدد الألوان المختلفة المتاحة فى نمط النصوص ؟ وما هو عدد المتاح فى نمط البيانات ؟
- ٤ - ١٢ ما هى النقطة ؟ وفيما تستخدم ؟
- ٥ - ١٢ ما الغرض من جملة SCREEN ؟ لخص القواعد التى تطبق لاستخدامها .
- ٦ - ١٢ ما الغرض من جملة PSET ؟ لخص القواعد التى تطبق لاستخدامها .
- ٧ - ١٢ ما الغرض من جملة PRESET ؟ وكيف تختلف عن جملة PSET ؟ لخص القواعد التى تطبق لاستخدامها .
- ٨ - ١٢ كيف يمكن توليد نقط مفردة فى نسختك من البيسك ؟
- ٩ - ١٢ ما الغرض من جملة LINE ؟ لخص القواعد التى تطبق لاستخدامها .
- ١٠ - ١٢ كيف يمكن توليد خط مفرد فى نسختك من البيسك ؟
- ١١ - ١٢ اشرح كيف ان عرضا على الشاشة يتكون كليا من نقط فردية أو خطوط (ليست أيشيكال) يمكن أن ينشط ، أى كيف يمكن أن تعرض نقط أو خطوط يمكن أن تقرر وفيما بعد تسمح مثلما فى برنامج (الفن الحركى) .
- ١٢ - ١٢ كيف يمكن لجملة LINE أن تستخدم لتوليد مستطيل مغلق ؟ كيف يمكن لهذا المستطيل أن يُملأ بلون معين ؟
- ١٣ - ١٢ ما الغرض من جملة ؟ لخص القواعد التى تطبق لاستخدامها ؟.
- ١٤ - ١٢ ما الغرض من جملة PAINT ؟ لخص القواعد التى تطبق لاستخدامها .
- ١٥ - ١٢ كيف يمكن لجملتى CIRCLE و PAINT أن تستخدم لتوليد شكل دائرى يملأ بلون معين ؟
- ١٦ - ١٢ هل يمكن توليد دوائر مغلقة فى نسختك من البيسك ؟ اذا كان كذلك فكيف يتم ؟
- ١٧ - ١٢ كيف يمكن لجملة CIRCLE أن تستخدم لتوليد أقواس بدلاً الدوائر الكاملة ؟ كيف يمكن لهذه الأقواس أن تتصل بنقطة الأصل لتكون أشكال دائرية ؟
- ١٨ - ١٢ كيف يمكن لجملة CIRCLE أن تستخدم لتوليد قطاعات ناقصة ؟ كيف يمكن تحديد حجم واتجاه (أى أفقى أو رأسى) لقطاع ناقص ؟
- ١٩ - ١٢ هل يمكن توليد قطاعات ناقصة مغلقة فى نسختك من البيسك ؟ اذا كان كذلك كيف يمكن أن يتم ؟
- ٢٠ - ١٢ اشرح كيف ينشط نص معروض على الشاشة فى الاتجاه الأفقى كما اخل فراغ ثابت محدد ؟ .
- ٢١ - ١٢ كيف يمكن تحريك شكل بسيط فى نسختك من البيسك ؟
- ٢٢ - ١٢ ما الغرض من جملة GET عندما تستخدم فى وسيلة البيانات ؟ لخص القواعد التى تطبق لاستخدامها
- ٢٣ - ١٢ ما الغرض من جملة PYT عندما تستخدم فى وسيلة البيانات ؟ لخص القواعد التى تطبق لاستخدامها .

- ١٢- ٢٤ عند تكوين رسوم متحركة ، ما المميزات في استخدام جملتي GET و PUT أكثر من ببساطة توليد لشكل بالوان تبادلية (مختلفة) (أولاً بلون الواجهة ثم بلون الخلفية ؟)
- ١٢- ٢٥ ما المقصود ببيانات الحروف ؟ كيف يمكن مقارنة نوعيات بيانات الحروف مع أنواع أخرى ؟
- ١٢- ٢٦ هل تحتوي نسختك من البيسك على بيانات حروف خاصة ؟ إذا كانت كذلك فما هي ؟ وماذا يكون كود ASCII لها ؟

مسائل تكميلية

Supplementary Problems

المسائل التالية تختص بجمع المعلومات بدلاً من الحل الحقيقي للمسائل . اجب عن الأسئلة باستخدام النسخة الخاصة بك من بيسك الحاسب الدقيق .

- ١٢- ٢٧ هل حاسبك الدقيق يحتوي على امكانية عروض البيانات ؟ اذا كان كذلك ، هل يمكن لعرض بيانات أن يتم تكوينها داخل برنامج البيسك ؟ كم عدد أنماط البيانات المتاحة ؟ ما مستوى التحليل المتاح مع كل أسلوب ؟
- ١٢- ٢٨ هل اللون متاح ؟ اذا كان كذلك ما هو عدد الالوان التي يمكن عرضها في أى وقت واحد داخل اسلوب البيانات ؟ ما هي هذه الألوان ؟ هل هناك تمييز بين الوان الواجهة والوان الخلفية ؟
- ١٢- ٢٩ حدد كيف يمكن تنفيذ الخصائص التالية في البيسك ؟
 (أ) كيف يمكن التوصل إلى كل اسلوب في البيانات ؟
 (ب) كيف يمكن تحديد لون مفرد ؟
 (ج) كيف يمكن توليد نقط مفردة ؟ وكيف يمكن مسحها ؟
 (د) كيف يمكن توليد خطوط ؟
 (هـ) هل يمكن توليد مستطيلات بجملة واحدة ؟ واذا كان كذلك فهل يمكن ملء مستطيلات بلون مصمت ؟
 (و) هل يمكن توليد دوائر بجملة واحدة ، واذا كان كذلك فهل يمكن ملء الدوائر بلون مصمت ؟
 (ز) هل يمكن توليد قطاعات ناقصة بجملة مفردة ؟ واذا كان كذلك فهل يمكن ملء هذه القطاعات بلون مصمت ؟
 (ح) هل يمكن توليد اقواس دائرية أو قطاعات ناقصة بجملة واحدة ؟ .
 (ط) هل هناك جمل خاصة متاحة لعمل رسوم متحركة ؟ واذا كان كذلك فما هي ؟ وكيف تعمل ؟
 (ى) هل يمكن توليد نص من خلال اسلوب البيانات ؟
 (ك) هل هناك خصائص خاصه لبيانات لم يتم شرحها في هذا الفصل ؟ فاذا كانت كذلك ، فما هي ؟ وكيف تعمل ؟
 (ل) هل يحتوي حاسبك الدقيق على حروف بيانية خاصة ؟ اذا كان كذلك ، فما هي وكيف يمكن التوصل لهذه الحروف في البيسك ؟

مسائل للبرمجة Programming Problems

- ١٢ - ٣٠ عدل البرامج الآتية بحيث يمكن تغيير لوحة الألوان ولون الخلفية في أى وقت بضغط مفاتيح دالة مناسبة (مثل F1 لتغيير لوحة الألوان و F2 لتغيير لون الخلفية) .
(أ) نقط الفراغ (مثال ١٢ - ٨) .
(ب) الخطوط المتحركة (الفن الحركي) (مثال ١٢ - ١١) .
(ج) المشاكل (مثال ١٢ - ١٦)
(د) الدوائر المتعددة (مثال ١٢ - ١٨)
- ١٢ - ٣١ غير بعض القيم العددية التى يحتوتها برنامج الفن الحركي فى مثال ١٢ - ١١ . جرب مع قيم مختلفة حتى تجد فئة من القيم التى ترغب فيها تماما .
- ١٢ - ٣٢ عدل البرنامج فى مثال ١٢ - ١٦ (المشاكل) بحيث يملأ الشاشة بعدد أكبر من المستطيلات . تأكد من اختزال حجم المستطيلات للتعويض عن الأعداد الأكبر عن العدد الكبير .
- ١٢ - ٣٣ عدل البرنامج فى مثال ١٢ - ١٦ (المشاكل) بحيث تنتشر الاحداثيات المولدة عشوائيا بانتظام على الحرف كله بدلا من أن تكون متجمعة بالقرب من مركز الشاشة . ماهو التأثير الذى تفضله ؟
- ١٢ - ٣٤ عدل البرنامج فى مثال ١٢ - ١٢ (الانحدار الخطى مع عرض بياني) بحيث أن محورى X و Y يوضع لها عناوين ضمن البرنامج مما يسمح لتوليد عنوان بالقرب من أعلى الشكل البياني .
- ١٢ - ٣٥ تم توسيع البرنامج فى مثال ١٢ - ١٢ بحيث يمكن توفير معادلات أسية وكثيرات الحدود لمجموعة من البيانات المدخلة ثم رسمها ، إستخدام نقط قربه من بعضها لتثيل المنحنيات عند رسم الأشكال .
(مثال ٧ - ٢٢ يعطى المعادلات المناسبة لتوفيق كل نوع من المنحنيات على اساس طريقة المربعات الصغرى) . ضمن قائمه التى تسمح للمستخدم أن يختار نوع المنحنى المطلوب . تأكد من تضمين ملفقات مناسبة واختبارات خطأ لكل من البيانات العددية واختبارات القائمة .
- ١٢ - ٣٦ عدل مولد الخريطة الدائرية فى مثال ١٢ - ٢٤ بحيث يملأ كل قطعه دائرية بلون غير لون الخلفية . تأكد من أن القطع المتجاورة ليس لها نفس اللون .
- ١٢ - ٣٧ عدل مولد خريطة الأعمدة فى مثال ١٢ - ٣٢ بحيث يمكن عرض مجموعة واحدة من البيانات المختلفة أو مجموعتين أو ثلاث فى نفس الوقت بشرط أن كل مجموعات البيانات يكون لها نفس العدد من الأعمدة . ضع الأعمدة المتناظرة واحدة بعد الأخرى بدون فراغات (أى ضع أول عمود لكل مجموعه بيانات فى تجميعه واحده ، ثم العمود الثانى لكل مجموعة بيانات .. الخ . إستخدام لون مختلف أو نموذج مختلف لكل مجموعة بيانات .
- ١٢ - ٣٨ أكتب برنامج ييسك للحاسب الدقيق الذى سوف يسمح لمجموعة من البيانات أن تدخل إلى الحاسب وعندئذ سوف يولد إما خريطة دائرية أو خريطة أعمدة (أنظر مثالى ١٢ - ٢٤ و ١٢ - ٣٢) ضمن قائمه التى تسمح للمستخدم أن يختار النوع المطلوب من الأشكال . تأكد من تضمين تلقينات مناسبة واختبارات خطأ لكل من البيانات العددية واختبارات القائمة .
- ١٢ - ١٩ عدل البرنامج فى مثال ١٢ - ٢٧ « بالون مع نص متحرك » بحيث أن المستخدم يمكن أن يحدد الرسالة عبر منطقة العرض . ضمن تلقين للعبارة المطلوبة عند بداية البرنامج قبل رسم البالون .
- ١٢ - ٤٠ غير بعض القيم العددية الموجودة فى برنامج الكرة المرتدة فى مثال ١٢ - ٢٩ . جرب باستخدام قيم مختلفة . حتى تجد مجموعة من القيم التى ترعيا جيدا . قارن عمل البرنامج مع الذى حصلت عليه بإستخدام القيم الأصلية .
- ١٢ - ٤١ عدل لعبة الكره والمضرب المعطى فى مثال ١٢ - ٣٠ بحيث يمكن للمستخدم أن يعين درجة الصعوبة قبل بداية اللعبة . وحتى تكون اللعبة أصعب اجعل المضرب أصغر ودع الكرة تتحرك أسرع . ضمن قائمة تلقينات مناسبة واختبارات الخطأ لتساعد المستخدم فى اختيار الدرجة المطلوبه من الصعوبة .

١٢ - ٤٢ بتوسيع البرنامج المعطى فى مثال ١٠ - ١٣ (برجه عصا التوجيه لتوليد بيانات الحروف) بحيث يمكن استخدام بيانات مختلفة للحروف . ضمن قائمة تسمح للمستخدم أن يختار الحروف البيانية . اسمح أيضا للمستخدم أن يغير لون الخلفية أو لون كل حرف جديد بالضغط على مفتاح داله مناسب . ضمن اختيار مفتاح الدالة فى القائمة . (اقتراح : احجز اقصى ٢٠ عمود يمين من الشاشة للقائمة . اسمح للباقي من الشاشة أن يستخدم كمنطقة البيانات) .

١٢ - ٤٣ حل مسألة ١٢ - ٤٢ بإستخدام متجول بدلا من عصا التوجيه .

١٢ - ٤٤ اكتب برنامج يسك للحاسب الدقيق ، مشابه للموجود فى مثال ١٠ - ١٣ والذي يسمح لعصا توجيه أو متجول أن بتوليد . نقط فردية فى اسلوب البيانات . ضمن البرنامج مايسمح لتغيير لون الخلفية أو لون كل نقطه جديده بضغط مفتاح دالة مناسب . عقد قائمة صغيرة عند قاع الشاشة لشرح استخدام مفاتيح الدوال .

١٢ - ٤٥ قم بتوسيع البرنامج فى مسألة ١٢ - ٤٤ بحيث يمكن رسم خطوط ومستطيلات ودوائر بالإضافة للنقط الفرديه . ضمن البرنامج املا كل شكل مصمت بلون والذي يمكن للمستخدم أن يختار من القائمة . أيضا اسمح للمستخدم أن يغير لون الخلفية أو لون كل شكل جديد بضغط مفتاح دالة مناسب .

١٢ - ٤٦ اكتب برنامج يسك للحاسب الدقيق والذي سوف يولد عرض شاشه كامله لشعار مدرستك أو رمز الشركه . ضمن لون إذا كان ذلك متاح .

١٢ - ٤٧ اكتب برنامج يسك للحاسب الدقيق والذي سوف يولد عرض شاشه كامله لعلم كلا من الدوال الآتية (بقائمة لكى تزيد الصعوبة) .

(هـ) الولايات المتحدة الأمريكية

(أ) اليابان

(و) المملكة المتحدة

(ب) فرنسا

(ى) كندا

(جـ) الدنمارك

(ز) المملكة العربية السعودية

(د) النرويج

١٢ - ٤٨ اكتب برنامج يسك للحاسب الدقيق والذي سوف يولد محور X ومحور Y مقسماً بذلك الشاشة إلى اربع ارباع متساوية . عندئذ إعرض شكل بياني للمعادلة

$$y = c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^4$$

مستخدما قيم من اختيارك الخاص لـ c_1, c_2, c_3, c_4, c_5 (لاحظ انه من الممكن أن نرسم خطوط مستقيمة ومعادلات درجة ثانية وثالثة .. إلخ . بوضع عدد معين من هذه الثوابت وبالصففر) .

اكتب البرنامج بطريقة معينة بحيث يمكن أن ينفذ بالتكرار ، بقيم مختلفة لـ c_1, c_2, c_3 و c_5 التى يمكن ادخالها من لوحة المفاتيح عند بداية كل تنفيذ .

١٢ - ٤٩ اكتب برنامج يسك للحاسب الدقيق والذي سوف يولد عرض بياني للمعادلة

$$y = 2e^{-0.1x} \sin(0.5x + c)$$

لقيم X تتغير من 0 إلى 60 ، و $C = 0$ ضمن محورى X و Y فى العرض البياني . ضع عنوان لكل محور .

١٢ - ٥٠ قم بتوسيع مسألة ١٢ - ٤٩ بحيث يحدث توليد العرض البياني داخل الحلقة التكرارية مع قيم (متزايدة) مختلفة مخصصة لـ K أثناء كل منسار .

١٢ - ٥١ هل يمكن توليد عدد من القروض البيانية المشوفة بالمعادلة التي تمثل حلزون أرخميدس وهي :

$$x = ar \cos r$$

$$y = br \sin r$$

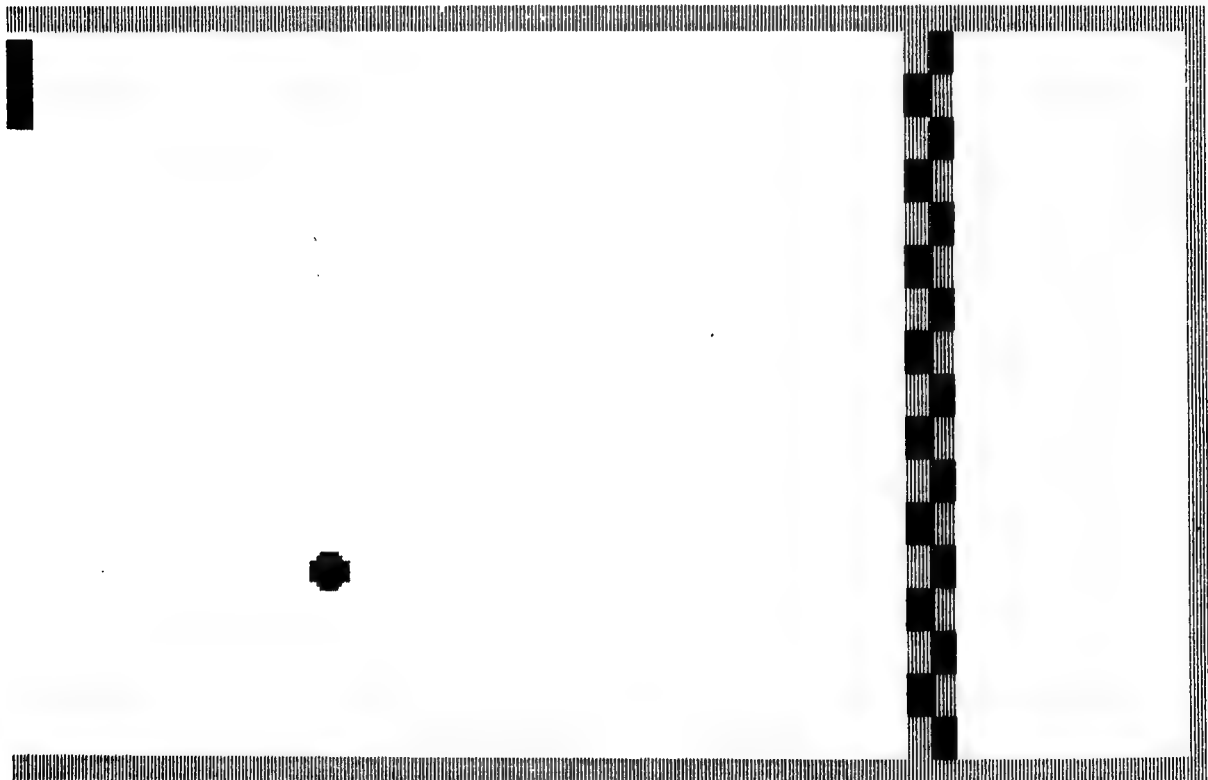
حيث a ، b ثوابت موجبة و r تمثل زاوية O بالتقدير الدائري .
اكتب برنامج يسك للحاسب الدقيق والذي سوف يولد عرض بياني لحلزون أرخميدس . اجعل الثوابت a ، b معالم مدخلة ولد سلسلة من القيم لـ r (وهكذا سلسلة من قيم x, y) بجعل هاتين المعادلتين في حلقة تكرارية FOR والتي تحتوى على معلمه STEP . ادخل قيمة معلمه STEP من لوحة المفاتيح ، برفقة قيمة a و b وعند بداية كل تنفيذ . (لاحظ أن كثير من العروض البيانية المختلفة يمكن توليدها بتعيين قيم مختلفة لـ a و b ومعلمه STEP .

١٢ - ٥٢ قم بتوسيع لعبة السرطان في مثال ٦ - ٢٠ ليحتوى عرض بياني للزهر بعد كل رمية . (لاحظ أن هذه المسألة يمكن أن تحل باستخدام أما بيانيات من نوع النقط أو بيانيات حروف) .

١٢ - ٥٣ اكتب برنامج يسك للحاسب الدقيق والذي سوف يسمح لشخص أن يلعب لعبة من تيك - تاك - تو أمام الكمبيوتر ضمن عرض يبين كلا من حركات اللاعبين مع محور X المألوف والصفري - (انظر مسألة ٦ - ١٥٢ ز)

١٢ - ٥٤ قم بتوسيع برنامج الروليت المشروح في مسألة ٦ - ٥٢ (ط) ليحتوى على عرض بياني لعجلة الروليت (مع اللون إذا كان ممكن) . بين متى تأتى الكرة (الزجاجية) لتقف بعد كل لفه .

١٢ - ٥٥ قم بتوسيع برنامج البيجو المشروح في مسألة ٦ - ٥٢ (ك) ليحتوى على عرض بياني لكارت بنجو الرئيسى . (هذا الكارت سوف يحتوى على ٧٥ توفيقه ممكنه لعدد الحروف الممكنة في خمس أعمدة ، معنون B11,N,G,O على الترتيب . أول عمود سوف يحوى الأرقام 1-15 ، والعمود الثانى سوف يحوى من 16-30 ... الخ) استخدم بيانيات الحروف لتوليد العرض . عرف كل توقيت لرقم الحرف كما سحبها (أى ظلل الموضع على الكارت أو غير اللون) .



١٢ — ٥٦ واحدة من أقدم ألعاب الفيديو والشعبية لعبة تسمى Brickout هذه اللعبة مشابهة للعبة مضرب التجديف المشروحة في مثال ١٢ — ٣٠ فيما عدا إنه يوجد « حائط مستطيل من المظلل » . موضوعه بالقرب من الجانب الأيمن لمنطقة اللعب ، كما تم مناقشته في شكل ١٢ — ٣٦ .

إذا ضربت الكرة طوبه ، ترتد الكرة كما هو معتاد ولكن الطوبه تختفي واللاعب يحصل على نقطة . إذا مرت الكرة خلال قناة تنشأ عن الطوب المفقود بلمس حائط مستطيل ، عندئذ الكرة يمكن أن تستمر لترتد خلال الجزء الأيمن لمنطقة اللعب تضغط وتنزع الكتلة حتى تمر مرة ثانية من خلال قناة للجزء الأيسر من منطقة اللعب . اللعب يستمر حتى يختفي الطوب أو يفقد اللاعب الكرة والمجداف . يسمح للاعب له بخمس كرات أثناء كل لعبة .

اكتب برنامج ييسر للحاسب الدقيق ليلعب لعبات متتالية من Brickout ، اجعل كل لعبة تالية أكثر صعوبة يجعل المضرب أصغر وجعل الكرة تتحرك بسرعة أكبر ابدأ درجة جديدة لكل لعبة ولكن حافظ على تسجيل لأعلى درجة .

إجابات لمسائل تكميلية مختارة .

- ٢٨-١ (أ) احسب مساحة مثل معروف القاعدة والارتفاع .
 (ب) احسب محيط مستطيل معروف الطول والعرض .
 (ج) احسب قيمة :

$$\begin{aligned}w &= u + v \\x &= u - v \\y &= uv \\z &= u/v\end{aligned}$$

حيث u و v معروفة القيم

(د) احسب قيمة :

$$y = 1 + x + \frac{x^2}{2} + \frac{x^3}{6}$$

حيث x معروفة القيمة

٢٩-١ (أ)

```
10 INPUT R
20 LET C=2*3.141593*R
30 PRINT R,C
40 END
```

يمكن كتابة الجملة 20 كالتالي :

```
20 LET C=6.283186*R
```

```
10 INPUT B,H
```

```
20 LET L=(B+2+L+2)^.5
```

```
30 PRINT B,H,L
```

```
40 END
```

(ب)

```
10 INPUT U,V
```

```
20 LET W=(U-V)/(U+V)
```

```
30 PRINT U,V,W
```

```
40 END
```

(ج)

```
10 INPUT X
```

```
20 LET Y=100*(1+X+2*X+2+3*X+3)
```

```
30 PRINT X,Y
```

```
40 END
```

(د)

٣٠-١ توجد الأخطاء التالية :

١ - يحتوى السطر الثانى على جملتين (يسمح بجملة واحدة فقط فى السطر) .

٢ - السطر الثالث لا يحتوى على الكلمة الدالة LET (بعض نسخ البيسك تسمح بذلك) .

٣ - أرقام الجمل المتعاقبة فى البرنامج لا تتزايد (يجب أن يكون رقم السطر الرابع أكبر من 35 ولكن أقل من 40) .

-7328500 or -7.3285E+6	(أ)	5	(أ) ٤٢-٢
2851 or 0.2851E+4	(ب)	8000 or 8E+3	(ب)
0.2851E+10 or 2851E+6, etc.	(ج)	-1.8033E-9	(ج)
-16752.47 or -1.675247E+4	(د)	0.33333333	(د)

- (أ) صحيحة
(ب) صحيحة
(ج) صحيحة
(د) لا يمكن أن يحتوي الأس على رقم عشري .
(هـ) صحيحة
(و) لا تسمح بالفواصل (,) .
(ز) الأس له قيمة كبيرة .
(ح) صحيحة
(ط) أرقام معنوية كثيرة جداً .
(ي) صحيحة
(ك) يجب أن يتبع الحرف E أس رقمي
(ل) الأس مكتوب بطريقة غير صحيحة (يجب أن يقرأ E-2)

- (أ) ٤٥-٢ صحيحة
(ب) صحيحة
(ج) طويلة جداً لبعض نسخ البيسك .
(د) صحيحة
(هـ) غير مسوح بعلامات الاقتباس .
(و) صحيحة

- (أ) ٤٦-٢ عددي (صحيح) .
(ب) عددي (صحيح) .
(ج) سلسلة حرفية (صحيحة) .
(د) لا يمكن أن يتبع علامة الدولار رقم صحيح .
(هـ) يجب أن يكون الحرف الأول حرفاً من الحروف الهجائية .
(و) لا تسمح بعض نسخ البيسك أن يتضمن المتغير الحرفي رقماً صحيحاً .
(ز) حروف كثيرة جداً .
(ح) أرقام صحيحة كثيرة جداً .
(ط) حروف هجائية كثيرة جداً .
(ي) سلسلة حرفية (صحيحة) .
(ك) يجب أن يكون الحرف الأول حرفاً من الحروف الهجائية .
(ل) يجب أن يكون ثاني حرف ، إن وجد ، رقماً صحيحاً أو علامة الدولار .
(م) عددي (صحيح) .
(ن) عددي (صحيح) .

$T \uparrow (N+1)$	(أ) ٤٧-٢
$(X+3) \uparrow (1/K)$	(ب)
$2*(A/B) \uparrow .33333333 \text{ or } 2*(A/B) \uparrow (1/3)$	(ج)
$1.87*(U+V)-5.088*(X/Y+2*Z+2)$	(د)
$1-X+X \uparrow 2/2-X \uparrow 3/6+X \uparrow 4/24-X \uparrow 5/120$	(هـ)
$(2*(P/Q) \uparrow (K-1))/((R-3*T) \uparrow (1/M))$	(و)
$(I+J-1) \uparrow 2/5 \text{ or } 0.2*(I+J-1) \uparrow 2$	(ز)
$((X1+X2) \uparrow M*(Y1+Y2) \uparrow N)/((X1/Y1) \uparrow (M+N)*(X2/Y2) \uparrow (M-N)) \uparrow (1/(M*N))$	(ح)

٤٨-٢ يجب أن يتم مسبقاً تحديد قيمة المتغير الذي يظهر على الجانب الأيمن من علامة التساوى في جملة LET بقيمة عددية أو حرفية مناسبة.

10 LET P=758.33	(أ) ٤٩-٢
20 LET B=A	(ب)
30 LET F\$="PITTSBURGH, PA."	(ج)
40 LET N\$=M\$	(د)
50 LET Y3=X/(A+B-C)	(هـ)
60 LET K=K-2	(و)
70 LET C5=2*C5	(ز)
80 LET B=C=(A+2+B+2) \uparrow .5	(ح)
10 LET W=((A+3)*B \uparrow N)/(2.7*(C-D/B)+1)	(أ) ٥٠-٢
20 LET F=((A/B) \uparrow N/(C-D) \uparrow M)/(D/(B-A) \uparrow (N+M)) \uparrow (1/(N+M))	(ب)
30 LET Y=(A1-A2*X+A3*X \uparrow 2-A4*X \uparrow 3+A5*X \uparrow 4)/(C1-C2*X+C3*X \uparrow 2-C4*X \uparrow 3)	(ج)
40 LET P=R*A*(1+R) \uparrow N/((1+R) \uparrow N-1)	(د)

10 LET W1=(A+3)*B \uparrow N	(أ) ٥١-٢
15 LET W2=2.7*(C-D/B)+1	
20 LET W=W1/W2	
50 LET F1=(A/B) \uparrow N/(C-D) \uparrow M	(ب)
55 LET F2=D/(B-A) \uparrow (N+M)	
60 LET F=(F1/F2) \uparrow (1/(N+M))	
100 LET Y1=A1-A2*X+A3*X \uparrow 2-A4*X \uparrow 3+A5*X \uparrow 4	(ج)
105 LET Y2=C1-C2*X+C3*X \uparrow 2-C4*X \uparrow 3	
110 LET Y=Y1/Y2	
200 LET Q=(1+R) \uparrow N	(د)
205 LET P=R*A*Q/(Q-1)	

$f = a + 2b/\sqrt{c}$	(أ) ٥٢-٢
$f = a + \sqrt{2b/c}$	(ب)
$f = (a+2) \sqrt{b/c}$	(ج)
$f = \sqrt{(a+2)b/c}$	(د)
$g = (pq/r)(s/t)$	(هـ)

٥٣-٢ إذا كانت (Y-Z) تمثل كمية سالبة ، إذن فسوف نواجه صعوبة حيث لا يمكن رفع الكمية السالبة إلى قوة كسرية في النسالة.

$$P=-(2+4)=-16$$

٥٤-٢

$$P=(-2)^4=16$$

٥٥-٢

10 INPUT A,B,C,M\$,N\$

(أ) ٥٦-٢

10 INPUT A,N\$,B

(ب)

15 INPUT M\$,C

(ج)

10 INPUT A

12 INPUT B

14 INPUT C

16 INPUT M\$

18 INPUT N\$

10 PRINT "ENTER VALUES FOR A,B,C,M\$ AND N\$";

(د)

20 INPUT A,B,C,M\$,N\$

10 PRINT "ENTER VALUES FOR A,B,C,M\$, AND N\$"

(هـ)

20 INPUT A,B,C,M\$,N\$

100 PRINT A,B,C,M\$,N\$

(و)

100 PRINT A,B;C;M\$;N\$

(ز)

120 PRINT A,B;C;

(ح)

200 PRINT A,B;C;(A+B+C)/3;(A*B*C)^(1/3);(A+2+B+2+C+2)^.5

(ط)

210 PRINT

220 PRINT M\$,,,N\$

300 PRINT "A=";A;"B=";B;"C=";C

(ي)

300 PRINT "A=";A;"B=";B;"C=";C

أو

500 PRINT "NAME: ";M\$

(ك)

510 PRINT

520 PRINT "SOCIAL SECURITY NUMBER: ";N\$

76.2E-6,27.5E-12,-1000

(أ) ٥٧-٢

7SHARON,GAIL

7.0000062,275E-10,-1000

يمكن طباعة أول سطر من البيانات أيضاً بالصورة التالية :

7-743.08,00987,SUSAN

(ب)

أو

7-74308E+3,987E-2,SUSAN

7"NEW YORK","CHICAGO","SAN FRANCISCO"

(ج)

(علامات الاقتباس مطلوبة حول NEW YORK و SAN FRANCISCO لأن كليهما يتضمن مسافة خالية ، أما في حالة CHICAGO ، فإن علامات الاقتباس تكون اختيارية) .

72770543,"DECEMBER 29, 1963",48.8E+9,"ELEVEN O' CLOCK"

(د)

(يجب أن تحصر السلاسل الحرفية بين علامات الاقتباس وذلك لوجود المسافات الخالية والفصلات (,)) .

6.20000E-6 2.75000E-11 -1000 2770543 -743.08 9.87000E-3
4.88000E+10

(أ) ٥٨-٢

6.20000E-6 2.75000E-11 -1000
9.87000E-3 4.88000E+10

(ب) 2770543 -743.08

6.17250E-6 5.67734E-5 -75286.7

(ج)

SHARON

DECEMBER 29, 1963

ELEVEN O' CLOCK

(د)


```

110 REM AVERAGING OF AIR POLLUTION DATA
250 REM BEGIN LOOP TO CALCULATE CUMULATIVE SUM
80 LET A=S/N 'CALCULATE AVERAGE VALUE
20 INPUT X,T 'READ A DATA POINT

```

(أ) ٥٩-٢
(ب)
(ج)
(د)

صحيحة (أ) ٦٠-٢

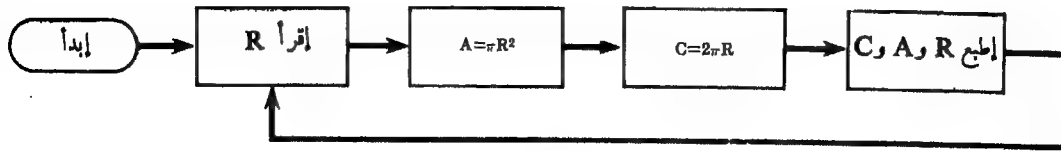
(ب) يجب أن يكون رقم الجسلة التي يتحول إليها التحكم رقماً صحيحاً موجباً وليس صيغة .

(ج) لا يمكن لجسلة GO TO أن تحول التحكم لنفسها .

(د) صحيحة

(هـ) لا يمكن لعلامات الاقتباس أن تظهر .

٦١-٢ خريطة سير العمليات مبينة في شكل المسألة ٦١-٢



شكل المسألة ٦١-٢

٦٢-٢

```

10 REM HELLO!
20 PRINT ,, "HELLO!"
30 END

```

(أ)

```

10 REM WELCOME!
20 PRINT "HI, WHAT'S YOUR NAME";
30 INPUT N$
40 PRINT
50 PRINT
60 PRINT "WELCOME "; N$; "!"
70 PRINT "LET'S BE FRIENDS!"
80 END

```

(ب)

```

10 REM TEMPERATURE CONVERSION PROBLEM
20 PRINT "TEMPERATURE IN DEGREES FAHRENHEIT=";
30 INPUT F
40 LET C=5*(F-32)/9
50 PRINT "DEGREES F=";F,"DEGREES C=";C
60 END

```

(أ) ٦٣-٢

```

10 REM PIGGY-BANK PROBLEM
20 PRINT "NUMBER OF HALF-DOLLARS=";
30 INPUT N1
40 PRINT "NUMBER OF QUARTERS=";
50 INPUT N2
60 PRINT "NUMBER OF DIMES=";
70 INPUT N3
80 PRINT "NUMBER OF NICKELS=";
90 INPUT N4
100 PRINT "NUMBER OF PENNIES=";
110 INPUT N5
120 LET S=.5*N1+.25*N2+.1*N3+.05*N4+.01*N5
130 PRINT "TOTAL AMOUNT OF MONEY=";S;" DOLLARS"
140 END

```

(ب)

لاحظ أن جملة END يمكن أن تستبدل بجملة GO TO ملائمة في كل من المسائل الموضحة عاليه . وسوف يسمح ذلك لكل برنامج بمعالجة مجموعات متعددة من البيانات بالتعاقب ..

٢-٦٤ فيما يلي برنامج يبسط كامل لكل مسألة ومع ذلك ، يجب أن يكون مفهوماً أن البرمجة الحقيقية لا يجب أن تبدأ حتى يتم تجهيز مخطط تمهيدي أو خريطة سير عمليات . كل برنامج مكتوب بطريقة تسمح بمعالجة مجموعات عديدة من البيانات في صورة متتالية .

```

10 REM VOLUME AND AREA OF A SPHERE
20 LET P=3.1415927
30 PRINT "RADIUS=";
40 INPUT R
50 LET V=4*P*R^3/3
60 LET A=4*P*R^2
70 PRINT "R=";R,"V=";V,"A=";A
80 PRINT
90 GO TO 30
100 END

```

(أ)

```

10 REM COMPUTATION OF MASS OF AIR IN A TIRE
20 PRINT "P=";
30 INPUT P
40 PRINT "V=";
50 INPUT V
60 PRINT "T=";
70 INPUT T
80 LET M=P*V/((.37*(T+460)))
90 PRINT "M=";M
100 PRINT
110 GO TO 20
120 END

```

(ب)

```

10 REM GEOMETRIC PROPERTIES OF A TRIANGLE
20 PRINT "A=";
30 INPUT A
40 PRINT "B=";
50 INPUT B
60 PRINT "C=";
70 INPUT C
80 LET S=(A+B+C)/2
90 LET A0=(S*(S-A)*(S-B)*(S-C))^0.5
100 LET R1=A0/S

```

(ج)

```

110 LET A1=3.14159*R1↑2
120 LET R2=A*B*C/(4*A0)
130 LET A2=3.14159*R2↑2
140 PRINT "AREA OF TRIANGLE=";A0
150 PRINT "AREA OF LARGEST INSCRIBED CIRCLE=";A1
160 PRINT "AREA OF SMALLEST CIRCUMSCRIBED CIRCLE=";A2
170 PRINT
180 GO TO 20
190 END

```

```

10 REM COMPOUND INTEREST PROBLEM (د)
20 PRINT "P=";
30 INPUT P
40 PRINT "I=";
50 INPUT I
60 PRINT "N=";
70 INPUT N
80 LET F=P*(1+I)↑N
90 PRINT "F="; F
100 PRINT
110 GO TO 20
120 END

```

إذا تضاعف الربح كل ربع سنة وليس سنوياً فيجب تغير الجملة رقم 80 لتقرأ كما يلي :

```
80 LET A=P*(1+I/4)↑(4*N)
```

```

10 REM GROWTH OF A BACTERIA POPULATION (هـ)
20 PRINT "T=";
30 INPUT T
40 LET C=.0289*T
50 LET F=1+C+C↑2/2+C↑3/6+C↑4/24+C↑5/120+C↑6/720+C↑7/5040+C↑8/40320
60 LET F=F+C↑9/362880
70 PRINT "P/P0=";F
80 PRINT
90 GO TO 20
100 END

```

لاحظ أن حساب حاصل الضرب المعامل (F) يتطلب جملتين حيث أن جملة واحدة يمكن أن تتعدى طول السطر (يتوقف ذلك على طولها) .

٣٢-٤ (أ) صحيحة

(ب) لا يمكن لهذا الشرط أن يستوفى أبداً .

(ج) صحيحة .

(د) صحيحة .

(هـ) صحيحة .

(و) لا يمكن مقارنة متغير رقمي بآخر حرفي .

(ز) صحيحة .

٤ - ٣٣ يجب أن تأتي السلسلة الحرفية المثلثة بواسطة P قبل السلسلة الحرفية المثلثة بواسطة Q وذلك كترتيب الهجائي .

٤ - ٣٤ (أ) صحيحة

(ب) غير صحيحة لغوياً وهيكلية (يمكن أن يتبع THEN رقم جملة فقط) .

(ج) لا تسمح كل نسخ البيسك باستخدام GO TO بدلا من THEN .

(د) صحيحة .

(هـ) يجب أن يكون رقم الجملة التي يتحول إليها التحكم رقماً صحيحاً موجباً وليس متغيراً .

(و) صحيحة .

(ز) لا يمكن أن يستوفى الشرط على الإطلاق .

40 IF K<15 THEN 50

٤ - ٣٥ (أ)

100 IF N\$="OPTION A" THEN 70

(ب)

110 GO TO 150

60 ...

(ج)

150 IF X>=100 THEN 200

160 LET J=J+1

170 INPUT X

180 GO TO 60

200 ...

20 ...

(د)

80 IF J=0 THEN 150

90 LET S=S+J

100 GO TO 20

150 ...

٤ - ٣٦ (أ) صحيحة

(ب) سوف تستمر الحلقة التكرارية إلى مالا نهاية لأن قيمة J سوف تكون دائماً 1 . (إذا تغيرت الجملة 80 إلى

80 GO TO 40

فسوف تنفذ الحلقة التكرارية بصورة صحيحة) .

(ج) سوف يتحول التحكم دائماً إلى الجملة رقم 200 .

(د) سوف تستمر الحلقة التكرارية إلى مالا نهاية ، حيث لن تتعدى قيمة X الرقم 100 على الإطلاق .

(هـ) صحيحة

٤ - ٣٧ (أ) لا يمكن أن يظهر متغير حرفي في جملة ON-GO TO .

(ب) صحيحة

(ج) لا يمكن جملة ON-GO TO أن تتحول التحكم لنفسها .

(د) يجب أن يكون رقم الجملة التي يتحول إليها التحكم رقماً صحيحاً موجباً . وليس متغيراً .

(هـ) صحيحة

(و) لا تسمح كل نسخ البيسك باستخدام THEN بدلا من GO TO .

(ز) غير صحيحة في الهيكل اللغوي (يجب أن تسبق ON الجملة GO TO)

- ٢٨ - ٤ (أ) سوف تنتج رسالة خطأ (لا يمكن أن نواصل التنفيذ لأن $J-K = -1$) .
 (ب) سوف يتحول التحكم إلى الجملة رقم 20 .
 (ج) سوف يتحول التحكم إلى الجملة رقم 20 .
 (د) سوف تنتج رسالة خطأ (لا يمكن أن نواصل التنفيذ لأن $J-K = 5$) .
 (هـ) سوف يتحول التحكم إلى الجملة رقم 100 .

- ٢٩ - ٤ (أ) صحيحة
 (ب) لا يمكن للصيغة الرياضية أن تظهر مكان المتغير الجارى .
 (ج) لا يمكن للمتغيرات الحرفية أن تظهر في جملة FOR-TO .
 (د) صحيحة (لاحظ أن $V(1)$ و $V(2)$ و $V(3)$ كلها متغيرات ذات أدلة ، حيث تمت مناقشتها في الفصل 5) .
 (هـ) صحيحة
 (و) جملة غير منطقية (مطلوب أن نقلل من قيمة المتغير الجارى حتى يتناقص ولكننا لانستطيع عمل ذلك لأن حجم الخطوة STEP قيمة موجبة) .

10 FOR I=1 TO 200
 ...
 70 NEXT I (أ) ٤٠ - ٤

توضع أرقام الجمل اختياريًا

10 FOR I=1 TO 200
 ...
 50 IF X<.001 THEN 175
 ...
 70 NEXT I
 ...
 175 ...
 10 FOR I=1 TO 73 STEP 3 (ب)
 ...
 70 NEXT I
 10 FOR C=.5 TO A+3-10 STEP A+B (د)
 ...
 70 NEXT C

- ٤١ - ٤ (أ) تتغير قيمة المتغير الجارى بداخل الحلقة التكرارية .
 (ب) صحيحة
 (ج) تتداخل الحلقات التكرارية بصورة متشابكة .
 (د) تستخدم الحلقة التكرارية الداخلية والخارجية نفس المتغير الجارى (X) . وكذلك فإن كلتا الحلقتين أيضاً تنهى عند جملة NEXT واحدة .
 (هـ) صحيحة طالما تم تحديد قيم عددية للمتغيرين T و T1 .
 (و) أن المتغير الجارى (C) في الجملة NEXT ليس هو نفس المتغير (X) في جملة FOR-TO .

10 LET Y=SQR(SIN(X)-COS(X)) (أ) ٤٠ - ٥
 10 LET P=Q*EXP(-Q*T) (ب)
 10 LET C=LOG(SQR(ABS(A+B)))+LOG(SQR(ABS(A-B))) (ج)
 10 LET W=ABS(ABS(U-V)-ABS(U+V)) (د)
 10 LET Z=COS(X+ATN(Y)) (هـ)

10 ON SGN((A*B-C*D)/(F+G))+2 GO TO 135,260,75 (أ) ٤١-٥
 100 PRINT TAB(4);"X=";X;TAB(28);"Y=";Y;TAB(52);"Z=";Z (ب)
 100 IF (N/2)=INT(N/2) THEN 200 (ج)

سوف يتحول التحكم إلى الجملة رقم 200 إذا كانت قيم N زوجية .

(د) سوف يستمر تحويل التحكم إلى الجملة رقم 200 إذا كانت قيم N زوجية .

(أ) ٤٢-٥ N\$ هي قائمة حرفية .
 (ب) A عبارة عن جدول رقمي و A\$ عبارة عن جدول حرفي و C قائمة رقمية و C\$ قائمة حرفية .
 (ج) P\$ قائمة حرفية و P جدول رقمي .
 (د) Z جدول رقمي .

(أ) ٤٣-٥ صحيحة

(ب) صحيحة

(ج) يجب أن تكون الكمية المحصورة بين القوسين في جملة DIM كمية صحيحة موجبة ، وغير مسموح باستعمال المتغيرات .

(د) لا يمكن أن تحتوي جملة DIM متغيرات بدون أدلة (في هذه الحالة المتغيرات C1 و C) .

(هـ) لا يمكن أن يكون الدليل قيمة سالبة .

(و) صحيحة

10 LET S=0 (أ) ٤٤-٥
 20 FOR I=1 TO 199 STEP 2
 30 LET S=S+X(I)^2
 40 NEXT I
 50 LET S1=SQR(S)

10 FOR I=1 TO 8 (ب)
 20 FOR J=1 TO 12
 30 LET H(I,J)=1/(I+J-1)
 40 NEXT J
 50 NEXT I

100 FOR I=1 TO N (ج)
 110 IF K(I)>15 THEN 130
 120 PRINT TAB(8);"I=";I;TAB(44);"K=";K(I)
 130 NEXT I

100 LET P=1 (د)
 110 FOR I=1 TO K
 120 LET P=P*W(I,I)
 130 NEXT I

100 FOR I=1 TO M (هـ)
 110 PRINT TAB(10);M\$(I,4)
 120 NEXT I

100 FOR I=1 TO M (و)
 110 PRINT M\$(I,4);" ";
 120 NEXT I

100 FOR J=1 TO N (ز)
 110 PRINT M\$(5,J);" ";
 120 NEXT J

```

10 READ L$(1),L$(2),L$(3),L$(4),P,Q,R,H$
20 READ T(1,1),T(1,2),T(1,3),T(1,4),T(2,1),T(2,2),T(2,3),T(2,4)
...
200 DATA WHITE, YELLOW, ORANGE, RED
210 DATA 2.25E+5,6.08E-9,-1.29E+12,RESTART
220 DATA 1,-3,5,-7,-2,4,-6,8

```

(أ) ٤٥-٥

يمكن لجمل READ و DATA أن تضم أو يتم توسيعها وذلك حسب الرغبة .

```

10 FOR I=1 TO 4
20   READ L$(I)
30 NEXT I
40 READ P,Q,R,H$
50 FOR I=1 TO 2
60   FOR J=1 TO 4
70     READ T(I,J)
80   NEXT J
90 NEXT I

```

(ب)

جمل DATA سوف تكون هي نفسها الموجودة في الجزء (أ) .

```

10 FOR I=1 TO 4
20   READ L$(I)
30 NEXT I
40 READ P,Q,R,H$
50 FOR I=1 TO 2
60   FOR J=1 TO 4
70     READ T(I,J)
80   NEXT J
90 NEXT I
...
150 RESTORE*
160 READ P1,Q1,R1

```

(ج)

جمل DATA سوف تكون هي نفسها كما في الجزء (أ) .

(د) الحل هو نفسه: الحل في الجزء (ج) ماعدا أن الجملة رقم 150 يجب أن تستبدل إما بالجملة :

```
150 RESTORE
```

أو بالجملة

```
150 RESTORE$
```

يجب أن تستبدل الجملة رقم 160 بالجملة :

```
160 READ A1$,A2$,A3$,A4$
```

```
10 DEF FNY(A,B,X)=A*X+B
```

(أ) ٤٦-٦

```
10 DEF FNQ(R)=C0+C1*R+C2*R^2+C3*R^3+C4*R^4
```

(ب)

يجب أن تعرف قيم المتغيرات C0 إلى C4 في مكان آخر من البرنامج ، قبل الرجوع والإشارة إلى الدالة .

```
10 DEF FNI(J,K)=(J+K)*(J+K)
```

(ج)

```
10 DEF FNR(A,B,C)
```

(د)

```
20 IF B^2<4*A*C THEN 50
```

```
30 LET FNR=SQR(B^2-4*A*C)
```

```
40 GO TO 60
```

```
50 LET FNR=SQR(4*A*C-B^2)
```

```
60 FNEND
```

```

10 DEF FNZ(Z)=INT(Z+.5)
10 DEF FNP(N)
20 LET P=1
30 FOR I=1 TO N
40   LET P=P*T(I)
50 NEXT I
60 LET FNP=P
70 FNEND

10 DEF FNR(A,B)
20 LET FNR=0
30 FOR I=1 TO 5
40   LET X=A+(B-A)*RND
50   IF X<=R THEN 70
60   LET FNR=X
70 NEXT I
80 FNEND

10 DEF FNW$(X)
20 ON SGN(X)+2 GO TO 30,50,70
30 LET FNW$="NEGATIVE"
40 GO TO 80
50 LET FNW$="ZERO"
60 GO TO 80
70 LET FNW$="POSITIVE"
80 FNEND

10 DEF FNL$(N$)
20 CHANGE N$ TO L
30 LET N=90
40 FOR I=1 TO L(0)
50   IF N<=L(I) THEN 70
60   LET N=L(I)
70 NEXT I
80 LET FNL$=CHR$(N)
90 FNEND

```

لاحظ أن المقدار الثابت 90 يساوى الحرف Z فى كود ASCII (أنظر الجدول ٦ - ١)

```

100 LET T=FNZ(C1+C2,3,X+Y)
100 LET Q=FNQ(LOG(X))
100 PRINT FNI(A-B,C)
100 LET D=ABS(X-FNZ(X))

10 FOR I=1 TO 60
20   IF FNP(I)>1000 THEN 40
30 NEXT I
40 PRINT "N"; I

10 FOR I=1 TO 20
20   PRINT "I=",I,"LARGEST R=",FNR(2,5)
30 NEXT I

```


- ٦-٤٤ (أ) عدد الخلاصات الموجودة في الإشارة للدالة لا يتوافق مع عدد الخلاصات الموجودة في تعريف الدالة .
 (ب) يحتوي هذا المثال على خطأين :
 (i) لم يعط اسم الدالة (FNC) قيمة من خلال الدالة .
 (ii) لا يمكن لجملة RETURN أن تظهر في تعريف الدالة .
 (ج) يحتوي هذا المثال على خطأين :
 (i) لا يمكن أن يتحول التحكم إلى البرنامج الصغير الفرعي بواسطة جملة GO TO .
 (ii) لا يمكن أن تظهر جملة FNEND في برنامج صغير فرعي .
 (د) يحتوي هذا المثال على خطأين :
 (i) أعط اسم غير صحيح للدالة
 (ii) لا يمكن أن تظهر الصيغ الرياضية كخلاصات زائفة .
 (هـ) الخلاصات غير موجودة في المرجع إلى الدالة .
 (و) لا يمكن تحويل التحكم إلى خارج برنامج صغير فرعي بواسطة جملة IF-THEN .
 (ز) منطق البرنامج غير صحيح . (تشير الدالة إلى البرنامج الصغير الفرعي ، وهو بدوره يشير إلى الدالة) .

- ٧-٣٩ (أ) صحيحة
 (ب) لا يمكن أن تظهر الصيغ الرياضية الخاصة بالمصفوفات في جمل المصفوفات .
 (ج) صحيحة .
 (د) صحيحة .
 (هـ) صحيحة .
 (و) لا يمكن تعديل المصفوفة بواسطة ضرب المصفوفة .
 (ز) لا يمكن القيام بعملية ضرب المصفوفة إلا إذا كان عدد أعمدة المصفوفة الأولى (X) هو نفسه عدد صفوف المصفوفة الثانية (Y) . وفي هذا المثال فإن المصفوفة (X) لها 20 عموداً ولكن المصفوفة (Y) لها عشرة صفوف فقط .
 (ح) يجب أن تكتب MAT INPUT بالصورة :
 50 MAT INPUT V
 (ط) صحيحة ، طالما يتم تحديد M و N بقيم صحيحة موجبة لا تتعدى 100 و 50 بالترتيب .
 (ي) لا يمكن استخدام خاصية البعد المتغير مع جملة MAT PRINT .
 (ك) صحيحة .
 (ل) يتمدد أحد الأبعاد الموصفة في جملة MAT ZER البعد المناظر له في جملة DIM .
 (م) لا يمكن الحصول على مقلوب مصفوفة غير مربعة .
 (ن) صحيحة .

٧-٤٠ (!)

```
100 MAT I=IDN
110 MAT B=TRN(A)
120 MAT C=B*A
130 MAT D=C-I
140 MAT F=(2*N+1)*D
```

سوف تكون F مصفوفة لها الأبعاد 10×10

```

100 MAT E=A*C
110 MAT F=B*D
120 MAT G=E-F
130 MAT H=INV(G)
140 LET D1=DET
150 PRINT "DETERMINANT OF G=";D.

```

(ب)

(ج) اجمع الجملة :

```

145 MAT PRINT G;H;

```

إلى الجزء (ب) السابق .

```

100 MAT A=ZER(12,8)

```

(د)

```

100 MAT READ A(8,12),B(6,15)

```

(هـ)

يجب أن تحتوي كتلة البيانات على عناصر المصفوفة ، بترتيب الصفوف A (أى الصف الأول ثم الثاني ... وهكذا) وتتبع هذه القيم قيم المصفوفة B بترتيب الصفوف أيضاً .

```

100 FOR I=1 TO 8
110   FOR J=1 TO 12
120     PRINT A(I,J);
130   NEXT J
140   PRINT
150 NEXT I
160 PRINT
170 FOR I=1 TO 6
180   FOR J=1 TO 15
190     PRINT B(I,J);
200   NEXT J
210   PRINT
220 NEXT I

```

(و)

الحلقات التكرارية FOR-TO مطلوبة لأن خاصية البعد المتغير غير متاحة مع جملة MAT PRINT .

٨ - ٣٤ (أ) تستخدم معظم نسخ البيسك جملة INPUT بدلا من جملة READ عند قراءة ملف بيانات متتال .

(ب) لا يمكن أن يحتوى ملف بيانات عشوائى على ثوابت حرفية ورقية معاً .

(ج) صحيحة طالما تم تحديد ملفات البيانات العشوائية لقنوات البيانات 1 و 2 .

(د) صحيحة .

(هـ) يجب أن تتضمن جمل SCRATCH و QUOTE أرقام قناة بيانات وليس أسماء ملفات .

(و) يجب أن تتضمن جملة FILES اسم ملف وليس متغيراً حرفياً .

(ز) لا يمكن وضع المؤشر وراء نطاق نهاية الملف .

(ح) لا يمكن كتابة سلسلة حرفية على ملف بيانات عشوائى.

```

10 FILES LIST1,,LIST2

```

(أ) ٨ - ٣٥

```

10 FILES NAMES$25,ACCTS%

```

(ب)

```

10 FILE :2,F$

```

(ج)

```

20 FILE :5,G$

```

```

NEW OR OLD-->NEW
NEW FILE NAME-->TAPE1
10 ...
20 ...
...
300 ...
SAVE
LIST

```

(د)

ملف البيانات 1 TAPE

```

NEW OR OLD-->NEW
NEW FILE NAME-->ITEMS
SAVE

```

(هـ)

```

OLD
OLD FILE NAME-->OLD1
SCRATCH
OLD
OLD FILE NAME-->OLD2
SCRATCH
OLD
OLD FILE NAME-->NEW1
RENAME OLD1
OLD
OLD FILE NAME-->NEW2
RENAME OLD2

```

(و)

(ز) محتمل أن البرنامج الموصوف في المسألة ٨ - ٣٥ (و) سوف يحتوى على الجملة :

```
10 FILES OLD1,OLD2,NEW1,NEW2
```

يجب أن تستبدل الجملة بسلسلة من الجمل التالية :

```

10 INPUT A$,B$,C$,D$
20 FILE #1,A$
30 FILE #2,B$
40 FILE #3,C$
50 FILE #4,D$

```

عند تنفيذ البرنامج فإن الأسماء الموجودة في ملفات الإدخال سوف تعطى للمتغيرات A\$ و B\$ ، أما الأسماء الموجودة في ملفات الإخراج فسوف تعطى للمتغيرات C\$ و D\$.

```

10 INPUT A$,B$
20 FILE #5,A$
30 FILE #3,B$
40 QUOTE #3
50 SCRATCH #3
60 INPUT #5,N,F$,X,Y,Z,G$
70 PRINT #3,N,Z,F$,G$
80 IF END #5, THEN 100
90 GO TO 60
100 END

```

(ح)

10 INPUT A\$,B\$	(ط)
20 FILE :5,A\$	
30 FILE :3,B\$	
40 READ :5,X	
50 WRITE :3,X	
60 IF LOC(5)=LOF(5) THEN 80	
70 GO TO 40	
80 END	
 50 INPUT P	(ع)
60 SET :6,P	
70 READ :6,X	
80 SET :2,P	
90 WRITE :2,X	
 100 LET P1=LOC(1)	(ك)
110 LET P4=LOC(4)	
120 IF P1=P4 THEN 200	
130 LET P2=P1	
140 IF P4<=P1 THEN 160	
150 LET P2=P4	
160 SET :2,P2	
 100 IF LOF(3)<>LOF(5) THEN 25	(ج)

المصطلحات العلمية (عربى - انجليزى)

(ا)

Procedure	اجراء
Logout procedure	اجراء التسجيل للخروج من النظام
Login procedure	اجراء التسجيل لدخول النظام
Iterative procedure	اجراء التكرار
Computational procedure	الاجراء الحسابى
Grammatical errors	الاطفاء اللغوية
Merge	ادمج
Bouncing ball	ارتداد كرة
Primary numbers	الارقام الاولى
Pseudo-random numbers	ارقام عشوائية كاذبة
Fibonacci numbers	ارقام فيبوناتسى
Significant figures	ارقام معنوية
Bouncing ball	ارتداد كرة
Integers	ارقام صحيحة
Reals	ارقام حقيقية
Mode	أسلوب
Modes of operation	اساليب التشغيل
Interpolation	الاستكمال
Initialize	استهل
Depreciation	استهلاك
Stock market	الاسواق المالية
Reference	الاشارة او الرجوع
Reordering a list	اعادة ترتيب قائمة
Virtually	افتراضيا
Debugging	اكتشاف الخطأ وتصحيحه
Blank spaces	أماكن خالية
Linear regression	انحدار خطى
Deviation	انحراف
Standard deviation	الانحراف المعيارى
Elementary	أولى

(ب)

Binary search

بحث ثنائى

In terms of	بدلالة
Source program	برنامج المنبع
Object program	برنامج الهدف
Structured program	برنامج هيكلى
Skeletal structure	بنشاء هيكلى
Items	بنود
High resolution graphics	بيانيات عالية الوضوح
Enhanced Basic	البيسك المحسن

(ت)

Spacing data items	تباعد بنود البيانات
Variance	التباين
Tracing	التتبع
Debugging	تحديد الأخطاء
Transform	تحويل
User-ID	تحقيق شخصية المستخدم
Transferring control	تحويل التحكم
Handling files	تداول الملفات
Hierarchy of operations	التدرج الهرمى للعمليات
Transpose	تدوير
Transpose of a matrix	تدوير مصفوفة
Compilation	ترجمة البرامج
Sequence	تسلسل
Error diagnostics	تشخيص الأخطاء
Processing a program	تشغيل برنامج
Manipulating files	التعامل مع الملفات
Instructions	تعليمات
Relational expression	تعبير مترابط
Successive substitution	التمويض المتتالى
Interpretation	تفسير البرامج
Encoding	تكويد شفرة
Prompting	تلقين
Air pollution	تلوث الهواء
Editing	تنقيح
Access	توصل
Direct access	التوصل المباشر
Curve fitting	توفيق منحنى

Generating random numbers

توليد ارقام عشوائية

(ج)

Table of variables

جدول من المتغيرات

Time sharing session

جلسة المشاركة الزمنية

(ح)

Microcomputers

الحاسبات الدقيقة

Commulative product

حاصل الضرب التراكمي

Data field

حقل بيانات

Nested loops

حلقات تكرارية متداخلة

Loop

حلقة تكرارية

(خ)

Actual arguments

خلاصات حقيقية

Dummy arguments

خلاصات زائفة

Argument

خلاصة

(د)

String function

دالة حروف

Statement function

دالة لها جملة واحدة

Multiline function

دالة متعددة الاسطر

Loop index

دليل الحلقة التكرارية

Built-in functions

دوال مبنية داخليا

Library functions

دوال مكتبة

Cycle

دورة

(ر)

Carriage return

رجوع العربية

Error message

رسالة خطأ

(س)

Data record

سجل بيانات

Horizontal velocity

السرعة الانغمية

String

سلسلة الحروف

(ش)

Monitor

شاشة عرض

(ص)

Hard-copy graph

صورة بيانية على ورق طباعة

Formula

صيغة رياضية

(ط)

Line printer	طابع السطور
Computational procedure	طريقة اجراء الحسابات
Overflow	طفح
Least squares method	طريقة اقل المربعات
Conversational manner	طريقة تخاطبية

(ع)

Quotation Marks	علامات الاقتباس
Colon (:)	علامة الوقف الاستدراكي
Customers	عملاء

(غ)

Unformatted	غير مصاغة
Unscrambling	غير مرئية

(ف)

Unique	فريد
Decoding	فك شفرة
Fibonacci	فيوناسي

(ق)

List of variables	قائمة من المتغيرات
Data Channel	قناة البيانات
Rules of the game	قواعد اللعبة
Initial value	القيمة الابتدائية
Final value	القيمة النهائية
Median	القيمة الوسيطة

(ك)

Data block	كتلة البيانات
Fraction	كسر
Key words	الكلمات الدالة
Console	كوتسول
Entity	كيان

(ل)

Hence	لذلك
Machine language	لغة الآلة
High level language	لغة رفيعة المستوى
Vertical scrolling	اللفة الرأسية

(م)

Programmer	مبرمج
Sequential	متتالية
Series	متسلسلة
Double precision	متضاعفة الدقة
Versatility	متعددة
Running variable	المتغير الجارى
Subscripted variables	متغيرات ذات ادلة
Calculated average	المتوسط المحسوب
Weighted average	المتوسط المرجح
Geometric average	المتوسط الهندسى
Array	مجموعة متراسة
Simulation	محاكاة
Trial-and-error	المحاولة والخطأ
Program outline	المخطط التمهيدى للبرنامج
Elimination scheme	مخطط الحذف
Graphical output	المخرجات البيانية
Inventory control	مراقبة المخزون
Weekly payroll	المرتبات الأسبوعية
Center of the screen	مركز الشاشة
Double Precision	مزدوجة الدقة
Area under the curve	المساحة تحت المنحنى
Inventory level	مستوى المخزون
formatted	مصاغة
Matrix	مصفوفة
Word processing	معالجة الكلمات
Relational operator	معامل الترابط
Parameters	معاملات
Logical operators	المعاملات المنطقية
Transaction	معاملة
Step value	مقدار الخطوة

Increment	مقدار الزيادة
Remark	الملحوظة
Associated	ملازم — مقترن
Random data file	ملف البيانات العشوائي
Sequential file	ملف متتابع
Suppliers	موردون
Uniformly distributed	موزع بانتظام
Normally distributed	موزع طبيعياً
Pointer	مؤشر
Cursor	مؤشر وامض
Location	موضع
Random number generator	مولد الرقم العشوائي

(ن)

Text	نص
Search interval	نطاق البحث
Terminal console	النهاية الطرفية المركزية
End of file	نهاية ملف

(هـ)

Histogram	هستوجرام
-----------	----------

(و)

Concatenation	الوصل
---------------	-------

(ى)

Truncate	يبتسر
Fluctuate	يتذبذب
Delete	يحذف
Assign	يحدد قيمة
Encode	يرمز أو يكود
Invoke	يستدعى
Decode	يفك الشفرة

Round-up

Cancel

Scratch

Utilize

Accomplish

يقرب

يلقى

يمسح

يستخدم

ينجز

الفهرس الأبجدي

- اكتشاف الأخطاء، ٦٨
آلة حاسبة مكتبية، ١٠٨
المعهد القومي الأمريكي للمعايير 19 ANSI
أمر AUTO، ٢٨١
أمر BYE، ٦٣، ٢٩٦
أمر CATALOG، ٦٥، ٢٨١
أمر CLEAR، ٢٨١
أمر DELETE، ٢٨١
أمر EDIT، ٢٨١، ٢٨٦
أمر GOOD BYE، ٦٥
أمر LIST، ٦٢، ٢٨١
أمر LOAD، ٢٨١ -
أمر NEW، ٥٨، ٢٨١
أمر NOTRACE، ٢٨١
أمر OLD، ٢٩٦
أمر RENAME، ٢٨١، ٢٩٤
أمر REPLACE، ٢٩٦
أمر RUN، ٥٩، ٢٨١
أمر SAVE، ٥٩، ٢٨١
أمر SCRATCH، ٢٩٦
أمر TRACE، ٢٨١
أمر UNSAVE، ٦٥
الانحدار الخطي، ٢٨٩
الانحراف، ١٣٦
فيما يتعلق بالمتوسط، ١٣٦
المعياري ١٣٨
أوامر التنقيح، ٦٥
أوامر النظام، ٦٥، ٢٧٩، ٢٩٦، ٣٠٨
أوامر نظام بيسك، ٦٥، ٢٧٩، ٣٠٨
إنجاد متوسط بيانات تلوث الهواء، ٨٩، ٣٠١
إنجاد متوسط درجات اختبار، ١٠٨، ١٣٦، ١٩٦، ٢٣٧،
٢٩٣، ٣٠٣
باسكال، ١٨
بايت BYTES، ١٣
التتبع، ٦٥
تحديد القيم لمتغيرات، ٩
التحكم في المخزون، ٢٤٧، ٢٩٤، ٣٠١
- إجراء التسجيل للخروج عن النظام، ٦٥
إجراء التسجيل لدخول النظام، ٥٨
إجراء التكرار، ٧٧، ١٤١
التشغيل التحويلي للحاسب، ١٦
الأخطاء اللغوية، ٦٦
المنطقية، ٦٧
إدخال برنامج، ٦٠
إدخال / إخراج المتجه والمصفوفة، ٢٠٣
الأدلة، ١١٣
الأرقام، ٢٧، ٢٧٢
أرقام ثنائية، ١٣
أرقام الجمل، ٢١
الأرقام العشوائية، ١٦٣
أرقام فيبوناتسي، البحث عن، ١٠٦
توليدها، ٩٤، ٢٧٧
الأرقام المعنوية، ٣٥، ٢٧٢
أساليب التشغيل، ١٤
مشاركة زمنية، ١٨
استخدام الأقواس، ٣٠
استدعاء البرنامج الفرعي، ١٦٩
الاستكمال بطريقة لأجرائي، ١٤٠
الأسلوب التخاطبي، ١٧
أسلوب التشغيل في دفعات ١٥
الإشارة إلى الدالة، ١٤٧
إعادة ترتيب قائمة من الأرقام ١٢٠ ٣٠٣
إعادة ترتيب قائمة من الأسماء، ١٣٥
إعادة قراءة البيانات، ١٢٧
الاعتماد على، ١٤
البت، ٨١، ٢٧٤
البحث عن الأرقام الأولية، ٩٤، ٢٧٧
البحث الثنائي، ٢٥٧، ٣٠٤

- البحث عن أقل قيمة ، ١٥٣
 البحث عن أقصى قيمة ، ١٤٩
 البحث في ملف بيانات ، ٢٥٧ ، ٣٠٤
 البرامج الفرعية المتداخلة ، ١٧٠
 برنامج بيسك ، بناءة ، ٢٠
 ترجمته ، ١٨
 كتابة برامج كاملة ، ٣٩
 البرمجة الهيكلية ، ٢٨٠
 برنامج الحاسب ، ١٢ ، ١٩
 المنبع ، ١٩
 الهدف ، ١٩
 برنامج المنبع ، ١٩
 برنامج الهدف ، ١٩
 بيانات الإخراج ، ١٢
 بيانات إدخال ، ١٢ ، ١١٨
 المصفوفة ، ٢٠٤
 بيانات ، حاسبات دقيقة ، ٢٨٦ ، ٢٩٢
 عالية الوضوح ، ٢٨٨
 منخفضة الوضوح ، ٢٨٦
 بيسك BASIC ، ١٨
 تاريخية ، ١٩
 تغييرات في ، ٢٠
 جمل ، ٢٠ ، ٢١
 الحاسب الدقيق ، ١٩ ، ٢٦٢
 الكلمات الدالة ، ٢٠
 مزاياه ، ٢١
 ميكروسوفت (البرامج الجاهزة الخاصة بالحاسبات الدقيقة)
 ٢٦١ ، ٢٦٩ ، ٢٧٩ ، ٢٨٥ ، ٢٩٥
 بيسك الحاسبات الدقيقة ، ١٩ ، ٢٦١
 PL/1 ، ١٨
 تاريخ الحاسب ، ١١
 التباين ، ١٣٢
 توليد علم أمريكي ، ١٨٩ ، ٢٨٧
 توماس كيرتز ، ١٩
 ثوابت ، ٢٦
 الجدول ، ١٠٨ ، ١٠٥ ، ٢٨١ ، ٣٠٧
 جدول من النوال ، ١٠٥ ، ٢٨١ ، ٣٠٧
 جنور المعادلات ، ٧٣ ، ١٠٣ ، ١٣٥ ، ١٤٧ ، ١٨٨ ، ٣٠١
 تحويل التحكم ، ٣٩ ، ٧١
 التدرج الهرمي للعمليات الحسابية ، ٢٨ ، ٢٦٦
 تدوير مصفوفة ، ٢٧
 ترتيب الكلمة المعثرة ، ١٠٩ ، ٣٠١
 الترجمة ، ١٨
 تشخيص الأخطاء ، ٦٣
 تشغيل برنامج ، ٥٩
 تشغيل درجات اختبار طالب ، ٢٣٦ ، ٢٩٣
 تصحيح الأخطاء ، ٥٨ ، ٦٥
 التعامل مع جدول ، ١١٨
 التعامل مع مصفوفة ، ١٩٥
 التعبيرات ، ٢٧
 قواعد خاصة ، ٢٩
 تعريف البرنامج الفرعي ، ١٦٢
 تعريف الدالة ، ١٤١
 تعليقات البرنامج ، ٣٩ ، ٢٩٧
 تغذية استمارة ، ٢٨٢
 التغير العشوائي ، ١٩٠
 التغير العشوائي ذو التوزيع العادي ، ١٩٠
 تغيير الأبعاد (جمل مصفوفة) ، ٢١٣
 التفرع ، غير المشروط ، ٣٩ ، ٦٧
 المتعدد ، ٧٧ ، ٢٦٩
 المشروط ، ٧١ ، ٧٢ ، ٢٦٨
 تفسير ، ١٨
 التقريب ، ٢٦٤
 تكاليف الرهونات ، ١٣٤ ، ١٣٥
 تكاليف رهونات منزل ، ١٣٤ ، ١٣٥
 التكامل العددي ، ١٣٧ ، ١٩٠
 التكرار ، ٧١ ، ٨٣ ، ٨٤
 التكرار المشروط ، ٢٧٠
 تكويد وفك نص ، ١٥٠ ، ١٩١
 التلقين ، ٥٧ ، ٢٧٠
 تنفيذ البرنامج تكرارياً ، ٤٠
 تنقيح سطر ، ٢٨٥
 توفيق منحني ، ٢١٤
 جملة IF-THEN ، ٧٦ ، ٢٧٩
 جملة IF-THEN-ELSE ، ٢٧٩
 جملة INKEY\$ ، ٢٨٢
 جملة INPUT ، ٣٣ ، ٢٨٢

- جملة INPUTS ، ٢٨٢
 جملة LET ، ٣١
 جملة LPRINT ، ٢٨٥
 جملة MAT CON ، ٢١١
 جملة MAT IDN ، ٢١١
 جملة MAT INPUT ، ٢٠٦
 جملة MAT INV ، ٢١٣
 جملة MAT PRINT ، ٢٠٣
 جملة MAT READ ، ٢٠٢
 جملة MAT TRN ، ٢١٢
 جملة MAT ZER ، ٢١٠
 جملة NEXT ، ٨٨
 جملة ON ERROR GO TO ، ٢٨٠
 جملة ON-GO-TO ، ٨١
 جملة ON-GOSUB ، ٢٨٠
 جملة PRINT ، ٣٤
 جملة PRINT USING ، ٢٨٣
 جملة PUT (بيانات) قبل مثال ١٢ — ٢٩
 جملة PUT ، ملف بيانات عشوائى — مثال ٩ — ٣١ ، ٢٩٧
 جملة RANDOMIZE ، ١٦٢
 جملة READ ، ١١٣
 جملة REM ، ٤٠
 جملة RESTORE ، ١٢٢
 جملة RESUME ، ٢٨١
 جملة RETURN ، ١٦٢
 جملة STOP ، ٧٨
 جملة TEXT ، ٢٨٦
 جملة VTAB ، ٢٨٣
 جملة WEND ، ٢٨١
 زائفة ، ١٤٢
 دالة ASC ، ١٥٢
 دالة CHR\$ ، ١٥٢
 دالة CVI ، ٢٩٤
 دالة INT ، ١٠٥
 دالة LEFT\$ ، ٢٨٨
 دالة LEN ، ٢٨٨
 دالة LOC ، ٢٤٣
 جنور المعادلات الجبرية ، ٧٣ ، ١٣٥ ، ١٤٧ ، ٢٠٩ ، ٣٠١
 جنور المعادلات التربيعية ، ٣٧ ، ٤٠ ، ١٠٣ ، ٣٠١
 جمع مصفوفة ، ١٩٣
 الجمل . ٢٠
 المتعددة ، ٢٦٦
 جمل (المصفوفة) MAT ، ١٩٢ ، ٢٩٦
 جمل المجاميع المتراصة الحرفية والمصفوفات ، ٢٢٤
 جمل مصفوفة ، ١٩٢
 جملة CALL ، ٢٩٦
 جملة CHANGE ، ١٥٦
 جملة CLS ، ٢٨٣ مثال ٣ — ١٠
 جملة COLOR ، ٢٨٦
 جملة DATA ، ١١٨
 جملة DEF ، ١٤٥
 جملة DEFDBL ، ٢٦٧ مثال ٩ — ١٢
 جملة DEFINT ، ٢٦٧ مثال ٩ — ١٢
 جملة DEFSNG ، ٢٦٧ مثال ٩ — ١٢
 جملة DEFSTR ، ٢٦٧ مثال ٩ — ١٢
 جملة DIM ، ١١٧
 جملة END ، ٣٨
 جملة FNEND ، ١٥٢
 جملة FOR-TO ، ٨٧
 جملة GET (بيانات) قبل مثال ١٢ — ٢٩
 جملة GET (ملف بيانات عشوائى) ٢٧٢ ، ٢٨٤ ، ٢٩٥
 جملة GO-TO ، ٤١
 جملة GOSUB ، ١٦٧
 جملة WHILE ، ٢٨١
 جملة WHILE ، ٢٨١
 جملة تحديد قيم لمصفوفة ، ١٩٢
 (جملة التعامل مع الملف) FILE ، ٢٤٩
 (جملة التعامل مع الملف) FILES ، ٢٣٣
 (جملة التعامل مع الملف) IFEND ، ٢٣٣
 (جملة التعامل مع الملف) INPUT ، ٢٣٣ ، ٢٩٣
 (جملة التعامل مع الملف) MARGIN ، ٢٥٤
 (جملة . نامل مع الملف) PAGE ، ٢٥٤
 (جملة التعامل مع الملف) PRINT ، ٢٣٦ ، ٢٩٣

- دالة LOR ، ٢٤٣ ، ٢٩٥
 دالة MIDS ، ٢٨٨
 دالة MKIS ، ٢٩٥
 دالة PEEK ، ٢٩٦
 دالة POS ، ٢٨٤
 دالة RIGHTS ، ٢٨٨
 دالة SPC ، ٢٨٣
 دالة SPC ، ٢٨٣
 دالة STR\$ ، ٢٨٨
 دالة TAB ، ١٠٥
 دالة USR ، ٢٩٦
 دالة VAL ، ٢٨٨
 الدوال ، جدول من ، ١٠٥ ، ٢٨٢ ، ٣٠٧
 العديدة الأسطر ، ١٤٧ ، ٢٩٦
 المكتبية ، ١٠٤ ، ٢٨٢
 الدوال الأولية ، ١٠٤
 الدوال العديدة الأسطر ، ٢٩٦
 الدوال القياسية ، ١٠٤ ، ٣٠٧
 الدوال المكتبية للغة البيسك ، ١٠٤ ، ٢٨٢ ، ٣٠٧
 الذاكرة ، ١٣
 الذاكرة المساعدة ، ١٤
 الربيع المركب ، ٥٣ ، ١٠٢ ، ١٣١ ، ٢٧٣ ، ٣٠١
 المسجلات ، ٢٩٤
 السرعة ، ١٤
 سرعة الحاسب ، ١٤
 سطر الأرقام ، ٢٠
 الكميات الحقيقية ، ٢٦٢
 الكميات الصحيحة ، ٢٦٢
 كميات متضاعفة الدقة ، ٢٦٢
 كوبول COBOL ، ١٨
 كود ASCIT ، ١٥٠
 لعبة BINGO ، ١٩١ ، ٣٠٢
 لعبة الأسواق المالية ، ٢١٤
 لعبة بلاك جاك ، ١٩٠
 لعبة تيك تاك تو ، ١٩٠ ، ٣٠٢
 لعبة حظ (لعبة الصدفة) ، ١٥٨ ، ١٨٨ ، ٣٠١
 لعبة الروليت ، ١٩١ ، ٣٠٢
 (جملة التعامل مع الملف) QUOTE ، ٢٣٦
 (جملة التعامل مع الملف) READ ، ٢٤٤
 (جملة التعامل مع الملف) RESET ، ٢٤٣
 (جملة التعامل مع الملف) RESTORE ، ٢٥٤
 (جملة التعامل مع الملف) SCRATCH ، ٢٣٦
 (جملة التعامل مع الملف) SET ، ٢٤٣
 (جملة التعامل مع الملف) WRITE ، ٢٤٦
 (جملة التعامل مع ملف مصفوفة) ، ٢٥٤
 جون كيمنى ، ١٩
 حاسبات كبيرة ، ١١
 حاسبات صغيرة ، ١١
 حاصل الضرب التراكمى ، ١٠١
 حساب الاستهلاك ، ٧٨ ، ٣٠١
 حساب المتوسط ، ١٠١
 المرجح ، ١٠١
 الهندسى ، ١٠١
 الحقول ، ٢٩٤
 حل المعادلات الآنية ، ٢٠٩ ، ٢١١
 حل المعادلات التفاضلية ، ١٣٧ ، ١٣٨
 كتابة كاملة ، ٣٧
 هيكل ، ٢٠
 حلول أرشميدس ، ٣٠٢
 الحلقات التكرارية المتداخلة ، ٨٨
 الخرج البيانى ، ١٧١ ، ٢٨٦ ، ٢٨٨ ، ٣٠١ ، ٣٠٢
 خريطة سير العمليات ، ١٣
 خصائص الحاسب ، ١١
 خلاصات ، ١٠٤
 السطر المتعدد الجمل ، ٢٦٥
 سلسلة حرفية مصاغة ، ٢٧١
 الصياغة ، ٢٧١
 الصيغ الرياضية ، ٢٧
 تقواعد خاصة ، ٢٩
 الضرب غير الموجه ، ١٩٤
 ضرب مصفوفة ، ١٣٢ ، ١٩٥
 طباعة مخرجات ، ٣٣

- لعبة كرابس ، ١٥٨ ، ١٨٨ ، ٣٠١
 لغة الآلة ، ١٨ ، ٢٩٥
 اللغة الرأسية ، ٢٩٥
 لوحة المفاتيح النهاية الطرفية للمشاركة الزمنية ، ١٥ ، ١٦٧
- المباعدة في السطر في جملة PRINT ، ٣٣
 المباعدة بين عناصر المخرجات في جملة PRINT ، ٣٥
 المتغير الجارى ، ٨٣ ، ٨٤
 المتغيرات ، ٢٧ ، ٢٦١
 الجارية ، ٨٣ ، ٨٤
 الحرفية ، ٢٧ ، ٢٦٢
 الحقيقة ، ٢٦٢
 ذات الأدلة ، ١٠٨
 الرقمية ، ٢٧
 الصحيحة ، ٢٦٢
 متضاعفة الدقة ، ٢٦٢
 مجموعات متراسة ، ١٠٨ ، ١١٢
 محاكاة ارتداد كرة ، ١٧٣ ، ١٨٨ ، ٣٠١
 مخزن وسيط ، ٢٩٥
 المرتبات ، الأسبوعية ، ١٨٨
 الشهرية ، ١٦٥
 المرتبات الأسبوعية ، ١٨٨
 المرتبات الشهرية ، ١٦٥
- ملفات البيانات ، ٢٣٢ ، ٢٩٢
 ملفات بيانات عشوائية ، ٢٤٢ ، ٢٩٤
 إنشاء ، ٢٤٩
 حاسب دقيق ، ٢٩٤
 قراءة ، ٢٤٤
 كتابة ، ٢٤٦
 مؤشر تحكم ، ٢٤٣
 ملفات البيانات المتتالية ، ٢٣٢ ، ٢٩٢
 إنشاء وتنقيح ، ٢٣٣ ، ٢٩٢
 قراءة ، ٢٣٤
 كتابة ، ٢٣٦
 المؤشر الوامض ، ٢٨٢
 مؤشرات ملف بيانات عشوائى ، ٢٤٣
 مواصفات وقت تشغيل ملف ، ٢٥٠
 مولد ييجلاتين ، ١٥٣ ، ٣٠١
 ميكروثانية ، ١٤
 ميكروسوفت بيسك ، ٢٦١ ، ٢٦٦ ، ٢٧٩ ، ٢٨٥ ، ٢٦٥
- طرح مصفوفة ، ١٩٤
 طريقة أقل المربعات ، ٢١٤ ، ٢١٥ ، ٢٨٩
 طريقة أولر ، ١٣٧
 المعدلة ، ١٧٤
 طريقة أولر المعدلة ، ١٧٤
 طريقة التعميض المتتالى ، ١٣٦
 طريقة رولج كوتا ، ١٣٧
 طريقة مونت كارلو ، ١٨٩
 طريقة نيوتن رابسون (طريقة نيوتن) ، ١٣٦
- الفراغات (المسافات الخالية) ، ٢٠
 فورتران FORTRAN ، ١٨
- قاعدة شبه المنحرف ، ١٣٦
 قانون سيمبسون ، ١٣٧
 قائمة أسماء وعناوين وأرقام تليفونات ، ٢٦٠ ، ٣٠٣
 قراءة بيانات الإدخال ، ٣١ ، ١٠٣ ، ١٢٣
 القوائم ، ١٠٨
- كثيرة الحدود (الجنادر) ، ١١١
 الكلمات ، ١٣
 كلمات الحاسب ، ١٣
 الكلمات الدالة ، ٢٠
 كلمة السر ، ٥٦
 مساحة دائرة ، ١٢ ، ١٥ ، ٢١ ، ٦١
 المشاركة الزمنية ، ١٦
 المشغل الدقيق ، ١١
 المصفوفات الخاصة ، ٢٠٥
 مصفوفة ، ١٩٢
 مصفوفة الوحدة ، ٢٠٦
- معالجة الكلمات ، ١٢
 معامل MOD ، ٢٦٤
 معامل قسمة رقم صحيح ، ٢٦٤
 المعاملات ، حسابية ، ٢٧ ، ٢٦٥
 مترابطة ، ٧١ ، ٢٦٥
 منطقية ، ٢٦٥
 معاملات الترابط ، ٧١ ، ٢٦٥
 المعاملات المنطقية ، ٢٦٥
 مفتاح ALTMODE ، ٥٩
 مفتاح BACKSPACE ، ٢٨٢
 مفتاح DELETE ، ٥٩

- مفتاح ESCAPE ، ٥٩
 مفتاح RUBOUT ، ٥٩
 مقلوب المصفوفة ، ٢٠٧
 ملاحظات البرنامج ، ٣٩
 الملحق ، نوع بيانات متغيرات رقمية ، ٢٦٣
 اسم ملف بيانات عشوائي ، ٢٤٢
 ملخص الجمل ، ٣٠٥
 ملخص جمل بيسك ، ٣٠٥
 ملفات ، ٦٢ ، ٢٣٢ ، ٢٩٢
 عشوائية ، ٢٤٢ ، ٢٩٤
 متتالية ، ٢٣٢ ، ٢٩٢
- نظرية الحد المركزي ، ١٩٠
 نقط صغيرة (بيكسلز) ، ٢٨٨
 النهاية الطرفية الذكية ، ٥٤
 النهاية الطرفية الذكية للحاسب ، ١٥ ، ٥٤
 النهاية الطرفية المركزية ، ١٥ ، ٥٤
- الوسيط ، ١٩٠
 الوصل ، ٢٦٦

ملحق (أ)

ملخص جمل BASIC القياسية

جمله	مثال	مراجع
CHANGE	10 CHANGE N\$ TO N	قسم ٦ - ٤
DATA	10 DATA 12,SEVENTEEN,-5	قسم ٥ - ٥
DEF	10 DEF FNR(A,B,C)=SQR(A↑2+B↑2+C↑2)	قسم ٦ - ١
DIM	10 DIM A(10,20),X(20),F\$(60)	قسم ٥ - ٤
END	10 END	قسم ٢ - ١١
FNEND	10 FNEND	قسم ٦ - ٢
FOR-TO	10 FOR J=1 TO 99 STEP 2	قسم ٥ - ٤
GO TO	10 GO TO 50	قسم ٢ - ١٤
GOSUB	10 GOSUB 300	قسم ٦ - ٩
IF-THEN	10 IF I>=100 THEN 80	قسم ٤ - ٢
INPUT	10 INPUT A,B,C,M\$,N\$	قسم ٢ - ٩
LET	10 LET A=3.141593*R↑2	قسم ٢ - ٨
NEXT	10 NEXT I	قسم ٤ - ٦
ON-GO TO	10 ON K GO TO 15,40,25,40,60	قسم ٤ - ٢
PRINT	10 PRINT "X=";X,"Y=";Y	قسم ٢ - ١٠
RANDOMIZE	10 RANDOMIZE	قسم ٦ - ٧
READ	10 READ K,N\$,Z(1)	قسم ٥ - ٥
REM	10 REM AREA OF A CIRCLE	قسم ٢ - ١٢
RESTORE	10 RESTORE	قسم ٥ - ٦
RETURN	10 RETURN	قسم ٦ - ٨
STOP	10 STOP	قسم ٤ - ٤
MAT =	10 MAT C=A	قسم ٧ - ١
MAT +	10 MAT C=A+B	قسم ٧ - ١
MAT -	10 MAT C=A-B	قسم ٧ - ١
MAT (K)*	10 MAT C=(10)*A	قسم ٧ - ١
MAT *	10 MAT C=A*B	قسم ٧ - ١
MAT CON	10 MAT B=CON	قسم ٧ - ٢
MAT IDN	10 MAT C=IDN	قسم ٧ - ٢
MAT INPUT	10 MAT INPUT A	قسم ٧ - ٢
MAT INV	10 MAT B=INV(A)	قسم ٧ - ٢
MAT PRINT	10 MAT PRINT A	قسم ٧ - ٢

جمله	مثال	مرجع
MAT READ	10 MAT READ A	قسم ٧ - ٧
MAT TRN	10 MAT B=TRN(A)	قسم ٧ - ٧
MAT ZER	10 MAT A=ZER	قسم ٧ - ٧
FILE	10 FILE :1,F\$	قسم ٨ - ٧
FILES	10 FILES SCORES	قسم ٨ - ١
IF END-THEN	10 IF END #1, THEN 130	قسم ٨ - ١
INPUT	10 INPUT #1,N,T\$,Y\$	قسم ٨ - ١
PRINT	10 PRINT #2,N;N\$	قسم ٨ - ١
QUOTE	10 QUOTE #2	قسم ٨ - ١
READ	10 READ :1,L	قسم ٨ - ٢
SCRATCH	10 SCRATCH #2	قسم ٨ - ١
SET (RESET)	10 SET :1,L	قسم ٨ - ٢
WRITE	10 WRITE :1,N	قسم ٨ - ٢

+	-	*	/	↑	المعاملات الرياضية
=	<>	<=	<	>=	>

ملحوظة : ↑ يظهر ك ^ على بعض أو في بعض الشاشات

ملحق (ب)

ملخص للدوال المكتبة القياسية في لغة البيسك

دالة	مثال	مرجع
ABS	10 LET Y=ABS(X)	قسم ٥ - ١
ATN	10 LET Y=ATN(X)	قسم ٥ - ١
ASC	10 LET N=ASC(T)	قسم ٦ - ٥
CHR\$	10 LET N\$=CHR\$(N)	قسم ٦ - ٥
COS	10 LET Y=COS(X)	قسم ٥ - ١
COT	10 LET Y=COT(X)	قسم ٥ - ١
DET	10 LET X=DET	قسم ٧ - ٢
EXP	10 LET Y=EXP(X)	قسم ٥ - ١
INT	10 LET Y=INT(X)	قسم ٥ - ١
LOC	10 LET N=LOC(1)	قسم ٨ - ٢
LOF	10 LET N1=LOF(3)	قسم ٨ - ٢
LOG	10 LET Y=LOG(X)	قسم ٥ - ١
NUM	10 LET N(0)=NUM	قسم ٧ - ٢
RND	10 LET X=RND	قسم ٦ - ٦
SGN	10 LET Y=SGN(X)	قسم ٥ - ١
SIN	10 LET Y=SIN(X)	قسم ٥ - ١
SQR	10 LET Y=SQR(X)	قسم ٥ - ١
TAB	10 PRINT TAB(N);X	قسم ٥ - ١
TAN	10 LET Y=TAN(X)	قسم ٥ - ١

ملحق (ج)

ملخص لأوامر النظام القياسية في لغة البيسك

الامر	الفرش
BYE	ينهى جلسة المشاركة الزمنية .
CATALOG	يعطى قائمة بأسماء كل الملفات التي تم الاحتفاظ بها .
GOODBYE	تبايا كابر BYE
LIST	يعطى قائمة بالملف الحالي .
NEW	يصف ان ملفا جديدا سوف ينشأ .
OLD	يتوصل الى ملف سبق الاحتفاظ به
RENAME	يسمح بتغير اسم الملف الحالي .
REPLACE	يتسبب الاحتفاظ بالملف الحالي (تخزينه) في مكان ملف سبق تخزينه بنفس الاسم . (سوف يلغى الملف القديم) .
RUN	يتسبب في ترجمة وتنفيذ البرنامج الحالي .
SAVE	يتسبب في الاحتفاظ بالملف الحالي (تخزينه)
SCRATCH	يزيل الملف الحالي من ذاكرة الحاسب .
SYSTEM	يهول التحكم من البيسك الى مراقب النظام .
UNSAVE	يلغى ملفا تم تخزينه في المخزن الدائم .

Summary of Microsoft BASIC

Statements

<i>Statement</i>	<i>Purpose</i>	<i>Example</i>
BEEP	Beeps the speaker	10 BEEP
BLOAD	Loads a binary memory image	10 BLOAD "SAMPLE"
BSAVE	Saves a binary memory image	10 BSAVE "SAMPLE",0,&H8000
CALL	Calls a machine language subroutine	10 CALL START
CHAIN	Passes control to another program	10 CHAIN "PROGRAM2"
CIRCLE	Generates circles, arcs and ellipses (graphics mode)	10 CIRCLE (160,100),30,2
CLOSE	Closes a file for input/output operations	10 CLOSE #1
CLS	Clears the screen and "homes" the cursor	10 CLS
COLOR	Sets colors or other screen attributes	10 COLOR 7,0,0
COMMON	Defines a common storage area, for passing variables to a chained program	10 COMMON A,B,C,T\$
DATA	Provides values for variables listed in READ statement	{ 10 READ A,B,C,T\$ 20 DATA 2.3,-0.1,6,RED
DATE\$	Sets the date	10 DATE\$="12/29/82"
DEF FN...	Defines a function	10 DEF FNA(X)=A*X^2+B
DEF(type)	Defines variable types (types can be INT, SNG, DBL or STR)	10 DEFINT I-N,X
DEF USR	Defines starting address for machine language subroutine	20 DEFSTR P 10 DEF USR=8000
DIM	Defines (dimension) arrays	10 DIM X(100),Z\$(20,100)
END	End of program	99 END
ERASE	Erases (eliminates) individual arrays	10 ERASE Z\$
ERROR	Simulates the occurrence of an error	{ 10 X=13 20 ERROR X
FIELD	Defines field length (random files)	10 FIELD 1,20 AS CUST\$
FOR and NEXT	Define the start and end of a FOR-TO loop	{ 10 FOR COUNT = 1 TO 100 60 NEXT COUNT
GET(text mode)	Reads a record from a random file to a memory buffer	{ 10 OPEN "R",#1,"DATA" 20 FIELD 1,20 AS CUST\$ 30 GET 1

<i>Statement</i>	<i>Purpose</i>	<i>Example</i>
GET(graphics mode)	Stores portion of graphics screen display in an array	10 GET (10,10)-(80,50),SHAPE
GOSUB	Transfers control to a subroutine	10 GOSUB 200
GOTO	Transfers control to a remote statement	10 GOTO 200
IF-THEN	Conditional execution	10 IF X>0 THEN 200
IF-THEN-ELSE	Conditional execution	10 IF X>0 THEN 200 ELSE X=0
INPUT	Enters data from the keyboard	10 INPUT A,B,C,T\$ 20 INPUT "X=",X
INPUT #	Enters data from a sequential file or device	10 OPEN "I",#1,"NAMES" 20 INPUT #1,N\$
KEY OFF	Turns off function key display	10 KEY OFF
KEY ON	Turns on function key display	10 KEY ON
KILL	Deletes an entire file	10 KILL "SAMPLE"
LET	Assignment statement (optional)	10 LET X=A+B+C
LINE	Generates lines and rectangles (graphics mode)	10 LINE (0,0)-(319,199) 20 LINE (10,5)-(50,80),2,BF
LINE INPUT	Reads an entire line from the keyboard, as a string	10 LINE INPUT T\$ 20 LINE INPUT;"Ans.";A\$
LINE INPUT #	Reads an entire line from a sequential file as a string	{ 10 OPEN "I",#1,"NAMES" 20 LINE INPUT #1,N\$
LOCATE	Specifies the current cursor position (row and column)	10 LOCATE 12,40
LPRINT	Prints data on the printer	10 LPRINT A,B,C,T\$ 20 LPRINT "X=";X
LPRINT USING	Prints formatted data on the printer	10 LPRINT USING "#.##";X
LSET	Places data into a random file buffer, left-justified	{ 10 OPEN "R",#1,"DATA" 20 FIELD 1,20 AS CUST\$ 30 LSET CUST\$=NAME\$
NAME	Renames a file	10 NAME "SAMPLE" AS "DATA"
ON ERROR GOTO	Transfers control if an error occurs	10 ON ERROR GOTO 500
ON-GOSUB	Transfers control to one of several subroutines	10 ON FLAG GOSUB 100,200,300
ON-GOTO	Transfers control to one of several destinations	10 ON K GOTO 80,120,160
ON KEY() GOSUB	Associates a function key with a subroutine	10 ON KEY(3) GOSUB 200 (refers to function key F3)
ON PEN GOSUB	Associates light pen activation with a subroutine	10 ON PEN GOSUB 300
ON STRIG() GOSUB	Associates a joystick button with a subroutine	10 ON STRIG(0) GOSUB 400
OPEN	Opens a file for input/output operations	10 OPEN "I",#1,"NAMES"

<i>Statement</i>	<i>Purpose</i>	<i>Example</i>
OUT	Sends a byte to an output port	10 OUT 127,3 (127 is the port no.)
PAINT	Fills an enclosed graphics shape with color	{ 10 CIRCLE (160,100),10,2 20 PAINT (160,100),2,2
PEN OFF	Turns off light pen read function	10 PEN OFF
PEN ON	Turns on light pen read function	10 PEN ON
POKE	Places a value in a specified memory location	10 POKE(32155,65) (memory location is 32155, value is 65)
PRESET	Erases a point (graphics mode)	10 PRESET (25,40)
PRINT	Displays data on the screen	10 PRINT A,B,C,T\$ 20 PRINT "X=";X
PRINT #	Writes data to a sequential file	10 PRINT #1,A;B;C;T\$
PRINT USING	Displays formatted data on the screen	10 PRINT USING "#.###";X
PRINT # USING	Writes formatted data to a sequential file	10 PRINT #1, USING "#.###";X
PSET	Generates a point (graphics mode)	10 PSET(25,40),1
PUT(text mode)	Writes a record from a memory buffer to a random file	10 PUT#1,22 (22 is the record no.)
PUT(graphics mode)	Displays graphics image stored in an array	10 PUT (120,80),SHAPE
RANDOMIZE	Initializes the random number generator	10 RANDOMIZE
READ	Assigns values in DATA statement to listed variables	{ 10 READ A,B,C,T\$ 20 DATA 2.3,-0.1,6,RED
REM	Places remarks in the program	10 REM *** PROGRAM 1 ***
RESTORE	Initializes the pointer in a DATA statement	{ 10 RESTORE 20 20 DATA 2.3,-0.1,6,RED
RESUME	Continues program execution after error correction routine	10 RESUME 100
RETURN	Used at end of subroutine; returns control to statement following GOSUB	{ 10 GOSUB 80 80 REM BEGIN SUBROUTINE 100 RETURN
RSET	Places data into a random file buffer, right-justified	{ 10 OPEN "R",#1,"DATA" 20 FIELD 1,20 AS CUST\$ 30 RSET CUST\$=NAME\$
SCREEN	Specifies current mode (text or graphics)	10 SCREEN 0
SOUND	Generates a sound with a fixed frequency and duration	10 SOUND 800,100 (800=frequency, 100=duration)

<i>Statement</i>	<i>Purpose</i>	<i>Example</i>
STOP	Terminates program execution	10 STOP
STRIG OFF	Deactivates joystick buttons	10 STRIG OFF
STRIG ON	Activates joystick buttons	10 STRIG ON
SWAP	Exchanges the values of two different variables	10 SWAP X,Y
TIME\$	Sets the current time	10 TIME\$="13:07:42"
WAIT	Suspends program execution until a specified bit pattern is detected in an input port	10 WAIT 16,6 (input port=16)
WHILE and WEND	Define the start and end of a conditional loop	10 COUNT=1 20 WHILE COUNT < 10 30 PRINT "COUNT=";COUNT 40 COUNT=COUNT+1 50 WEND
WIDTH	Specifies the number of characters per line	10 WIDTH 80
WRITE	Displays data on the screen (similar to PRINT)	10 WRITE A,B,C,T\$
WRITE #	Writes data to a sequential file (similar to PRINT #)	10 WRITE #1,A,B,C,T\$

Library Functions

<i>Function</i>	<i>Purpose</i>	<i>Example</i>
ABS	Returns absolute value	10 Y=ABS(X)
ASC	Returns ASCII code	10 Y=ASC(X\$)
ATN	Returns arctangent	10 Y=ATN(X)
CDBL	Converts to double precision	10 Y#=CDBL(X)
CHR\$	Returns character represented by given ASCII code	10 Y\$=CHR\$(X)
CINT	Converts to an integer	10 Y%=CINT(X)
COS	Returns the trigonometric cosine function	10 Y=COS(X)
CSNG	Converts to single precision	10 Y=CSNG(X#)
CSRLIN	Returns the vertical cursor position (line number)	10 Y=CSRLIN
CVD	Converts string to double precision value	10 Y#=CVD(X\$)
CVI	Converts string to integer value	10 Y%=CVI(X\$)
CVS	Converts string to real value	10 Y=CVS(X\$)
DATE\$	Returns the date	10 Y\$=DATE\$
EOF	Indicates an end-of-file	10 IF EOF(1) THEN 100
ERL	Returns the line number where an error occurred	(see next example)

<i>Function</i>	<i>Purpose</i>	<i>Example</i>
ERR	Returns an error code	10 PRINT ERR, ERL
EXP	Returns the exponential function	10 Y=EXP(X)
FIX	Converts to an integer (truncate)	10 Y%=FIX(X)
FRE	Returns the number of unused bytes of memory	10 Y=FRE(0)
HEX\$	Converts from decimal to hexadecimal	10 Y\$=HEX\$(X)
INKEY\$	Returns a character from the keyboard	10 Y\$=INKEY\$
INP	Returns a byte from an input port	10 Y=INP(127)
INPUT\$	Returns a multicharacter string from the keyboard	10 Y\$=INPUT\$(3)
INSTR	Returns the position where one string (X\$) is found within another string (T\$)	10 Y=INSTR(T\$,X\$)
INT	Returns the largest integer that does not exceed the specified value	10 Y%=INT(X)
LEFT\$	Returns the leftmost <i>n</i> characters of a string	10 Y\$=LEFT\$(X\$, 3)
LEN	Returns the number of characters in a string	10 Y=LEN(X\$)
LOC	Returns the current record number	10 Y=LOC(1)
LOF	Returns the file length, in bytes	10 Y=LOF(1)
LOG	Returns the natural logarithm	10 Y=LOG(X)
LPOS	Returns the column number of the current print-head position (for a printer)	10 IF LPOS>40 THEN PRINT "*"
MID\$	Returns an <i>n</i> -character string, starting at location <i>m</i>	10 Y\$=MID\$(X\$,5,3) (<i>m</i> =5, <i>n</i> =3)
MKD\$	Converts double-precision value to a string	10 Y\$=MKD\$(X#)
MKI\$	Converts integer value to a string	10 Y\$=MKI\$(X%)
MKS\$	Converts real value to a string	10 Y\$=MKS\$(X)
OCT\$	Converts from decimal to octal	10 Y\$=OCT\$(X)
PEN	Returns information associated with a light pen	10 X=PEN(1):Y=PEN(2)
PEEK	Returns the contents of a specified memory location	10 Y=PEEK(32155)
POINT	Returns the color of a point on the screen	10 Y=POINT(3,12)
POS	Returns the column number of the current cursor position	10 Y=POS(0)

<i>Function</i>	<i>Purpose</i>	<i>Example</i>
RIGHT\$	Returns the rightmost <i>n</i> characters of a string	10 Y\$=RIGHT\$(X\$, 3)
RND	Returns a random number between 0 and 1	10 Y=RND
SCREEN	Returns ASCII code for the character at the designated location	10 Y=SCREEN(5,12)
SGN	Returns an integer that indicates the sign of a value	10 Y=SGN(X)
SIN	Returns the trigonometric sine function	10 Y=SIN(X)
SPACES\$	Returns a sequence of blank spaces	10 PRINT X;SPACES\$(5);'
SPC	Generates blank spaces in a PRINT statement	10 PRINT X;SPC(5);Y
SQR	Returns the square root of a value	10 Y=SQR(X)
STICK	Returns joystick coordinates	10 X=STICK(0);Y=STICK(1)
STRIG	Returns information associated with joystick buttons	10 Y=STRIG(0)
STR\$	Converts a numerical value to a string	10 Y\$=STR\$(1000)
STRING\$	Returns an <i>n</i> -character string of repeated characters	10 Y\$=STRING\$(8,42) (<i>n</i> =8, ASCII char=42)
TAB	Tabs to a specified position in a print statement	10 PRINT X;TAB(18);Y
TAN	Returns the trigonometric tangent	10 Y=TAN(X)
TIMES\$	Returns the current time	10 Y\$=TIMES\$
USR	Accesses a machine-language subroutine	10 Y=USR(X)
VAL	Converts a string to a numerical value	10 Y=VAL(X\$)
VARPTR	Returns the memory address of a variable	10 Y=VARPTR(X)

System Commands

<i>Command</i>	<i>Purpose</i>	<i>Example</i>
AUTO	Automatic line numbering	AUTO 100,10
CLEAR	Clears values assigned to numeric and string variables	CLEAR
CONT	Resumes program execution after a break	CONT
DELETE	Deletes program lines	DELETE 180-230
EDIT	Accesses a line for editing	EDIT 100
FILES	Displays names of all files	FILES
KILL	Deletes an entire file	KILL "SAMPLE"
LIST	Lists the program, or parts of the program, on the screen	LIST
		LIST 100-160

<i>Command</i>	<i>Purpose</i>	<i>Example</i>
LLIST	Lists the program, or parts of the program, on a printer	LLIST LLIST 100-160
LOAD	Loads a program into memory	LOAD "SAMPLE"
MERGE	Merges a program file into the program now in memory	MERGE "TRIAL"
NAME	Renames a file	NAME "SAMPLE" AS "NEWPROG"
NEW	Deletes the program currently in memory	NEW
RENUM	Renumbers program lines automatically	RENUM
RESET	Closes all files and clears the memory buffer	RESET
RUN	Initiates program execution	RUN
SAVE	Saves the program currently in memory	SAVE "SAMPLE"
SYSTEM	Exits from BASIC to the operating system	SYSTEM
TRACE ON (TRON)	Activates tracing of program statements during program execution	TRACE ON TRON
TRACE OFF (TROFF)	Discontinues tracing of program statements during program execution	TRACE OFF TROFF

Operators (listed hierarchically)

<i>Operation</i>	<i>Operator</i>	<i>Operation</i>	<i>Operator</i>
1. Exponentiation	↑ or ^	8. Logical NOT	NOT
2. Negation	-	9. Logical AND	AND
3. Multiplication and division	* /	10. Logical OR	OR
4. Integer division	\	11. Logical XOR (exclusive OR)	XOR
5. Integer remainder	MOD	12. Logical EQV (equivalence)	EQV
6. Addition and subtraction	+ -	13. Logical IMP (implication)	IMP
7. Relational	= < > <= < >= >		

Other Punctuation

- Colon (:) Used to separate statements on the same line.
Example: 10 CLS : KEY OFF
- Apostrophe (') Used to designate comments on a statement line.
Example: 10 CLS 'clear the screen

NOTES:

1. Many specific implementations include additional commands.
2. Some statements can also be used as system commands (e.g., CLS).
3. Some statements or functions may have different interpretations or multiple interpretations (e.g., GET, PUT).

The ASCII Character Set

ASCII Value	Character	ASCII Value	Character	ASCII Value	Character	ASCII Value	Character
000	NUL	032	blank	064	@	096	`
001	SOH	033	!	065	A	097	a
002	STX	034	"	066	B	098	b
003	ETX	035	#	067	C	099	c
004	EOT	036	\$	068	D	100	d
005	ENQ	037	%	069	E	101	e
006	ACK	038	&	070	F	102	f
007	BEL	039	'	071	G	103	g
008	BS	040	(072	H	104	h
009	HT	041)	073	I	105	i
010	LF	042	*	074	J	106	j
011	VT	043	+	075	K	107	k
012	FF	044	,	076	L	108	l
013	CR	045	-	077	M	109	m
014	SO	046	.	078	N	110	n
015	SI	047	/	079	O	111	o
016	DLE	048	0	080	P	112	p
017	DC1	049	1	081	Q	113	q
018	DC2	050	2	082	R	114	r
019	DC3	051	3	083	S	115	s
020	DC4	052	4	084	T	116	t
021	NAK	053	5	085	U	117	u
022	SYN	054	6	086	V	118	v
023	ETB	055	7	087	W	119	w
024	CAN	056	8	088	X	120	x
025	EM	057	9	089	Y	121	y
026	SUB	058	:	090	Z	122	z
027	ESC	059	;	091	[123	{
028	FS	060	<	092	\	124	
029	GS	061	=	093]	125	}
030	RS	062	>	094	↑	126	~
031	US	063	?	095	—	127	DEL

Note: The first 32 characters and the last character are control characters; they cannot be printed.

رقم الإيداع ٨٩/٧٠٩٦

مطابع الكتب المصري الحديث
MODERN EGYPTIAN PRESS
الكتاب ١٠ الطريق الزراعي مصر الجديدة ص ب ٣٥ قسوة الجديدة ت ٩٥٧٩٥٩

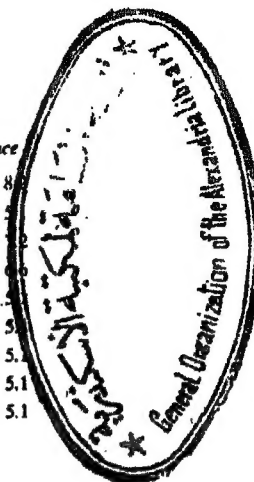
Summary of Standard BASIC Statements (Appendix A)

Statement	Example	Reference	Statement	Example	Reference
CHANGE	10 CHANGE N\$ TO N	Section 6.4	MAT =	10 MAT C=A	Section 7.1
DATA	10 DATA 12,SEVENTEEN,-5	Section 5.5	MAT +	10 MAT C=A+B	Section 7.1
DEF	10 DEF FNR(A,B,C)=SQR(A ² +B ² +C ²)	Section 6.1	MAT -	10 MAT C=A-B	Section 7.1
DIM	10 DIM A(10,20),X(20),F\$(60)	Section 5.4	MAT (K)*	10 MAT C=(10)*A	Section 7.1
END	10 END	Section 2.11	MAT *	10 MAT C=A*B	Section 7.1
FNEND	10 FNEND	Section 6.3	MAT CON	10 MAT B=CON	Section 7.3
FOR-TO	10 FOR J=1 TO 99 STEP 2	Section 4.5	MAT IDN	10 MAT C=IDN	Section 7.3
GO TO	10 GO TO 50	Section 2.14	MAT INPUT	10 MAT INPUT A	Section 7.2
GOSUB	10 GOSUB 300	Section 6.9	MAT INV	10 MAT B=INV(A)	Section 7.3
IF-THEN	10 IF I>=100 THEN 80	Section 4.2	MAT PRINT	10 MAT PRINT A	Section 7.2
INPUT	10 INPUT A,B,C,M\$,N\$	Section 2.9	MAT READ	10 MAT READ A	Section 7.2
LET	10 LET A=3.141593*R12	Section 2.8	MAT TRN	10 MAT B=TRN(A)	Section 7.3
NEXT	10 NEXT I	Section 4.6	MAT ZER	10 MAT A=ZER	Section 7.3
ON-GO TO	10 ON K GO TO 15,40,25,40,60	Section 4.3	FILE	10 FILE :1,F\$	Section 8.3
PRINT	10 PRINT "X=";X;"Y=";Y	Section 2.10	FILES	10 FILES SCORES	Section 8.1
RANDOMIZE	10 RANDOMIZE	Section 6.7	IF END-THEN	10 IF END #1, THEN 130	Section 8.1
READ	10 READ K,N\$,Z(1)	Section 5.5	INPUT	10 INPUT #1,N,T\$,Y\$	Section 8.1
REM	10 REM AREA OF A CIRCLE	Section 2.13	PRINT	10 PRINT #2,N;N\$	Section 8.1
RESTORE	10 RESTORE	Section 5.6	QUOTE	10 QUOTE #2	Section 8.1
RETURN	10 RETURN	Section 6.8	READ	10 READ :1,L	Section 8.2
STOP	10 STOP	Section 4.4	SCRATCH	10 SCRATCH #2	Section 8.1
			SET (RESET)	10 SET :1,L	Section 8.2
			WRITE	10 WRITE :1,N	Section 8.2

Arithmetic Operators: + - * / ↑
 Relational Operators: = <> <= < >= >

Summary of Standard BASIC Library Functions (Appendix B)

Function	Example	Reference	Function	Example	Reference
ABS	10 LET Y=ABS(X)	Section 5.1	LOF	10 LET N1=LOF(3)	Section 8.1
ATN	10 LET Y=ATN(X)	Section 5.1	LOG	10 LET Y=LOG(X)	Section 5.1
ASC	10 LET N=ASC(T)	Section 6.5	NUM	10 LET N(0)=NUM	Section 6.5
CHR\$	10 LET N\$=CHR\$(N)	Section 6.5	RND	10 LET X=RND	Section 6.5
COS	10 LET Y=COS(X)	Section 5.1	SGN	10 LET Y=SGN(X)	Section 5.1
COT	10 LET Y=COT(X)	Section 5.1	SIN	10 LET Y=SIN(X)	Section 5.1
DET	10 LET X=DET	Section 7.3	SQR	10 LET Y=SQR(X)	Section 5.1
EXP	10 LET Y=EXP(X)	Section 5.1	TAB	10 PRINT TAB(N);X	Section 5.1
INT	10 LET Y=INT(X)	Section 5.1	TAN	10 LET Y=TAN(X)	Section 5.1
LOC	10 LET N=LOC(1)	Section 8.2			



Summary of Standard BASIC Systems Commands (Appendix C)

Command	Purpose	Command	Purpose
BYE	Terminates timesharing session.	REPLACE	Causes the current file to be saved (stored) in place of the file previously stored with the same name. (The old file will be deleted.)
CATALOG	Lists names of all files being saved.	RUN	Causes the current program to be compiled and executed.
GOODBYE	Same as BYE.	SAVE	Causes the current file to be saved (stored).
LIST	Produces a listing of the current file.	SCRATCH	Removes the current file from the computer's memory.
NEW	Specifies that a new file will be created.	SYSTEM	Transfers control from BASIC to the system monitor.
OLD	Accesses an existing file.	UNSAVE	Cancels permanent storage of a file.
RENAME	Allows the name of the current file to be changed.		

صدر أيضاً في الحاسبات للناشر

المدخل لعلم الحاسبات - بارف
الرياضيات الأساسية للحاسب (شوم) - ليستر
الخوارزميات والحاسبات الدقيقة - رولارد
الدوائر المتكاملة الرقمية وحاسبات - رولارد
استخدامات الخوارزميات الدقيقة - موريس
البرمجة بالفورتران (شوم) - ج. ليستر
البرمجة بلغة الباشكال - كير
* بحوث العمليات - برونسون
* البرمجة بلغة الباسك - جونفريد (إنجليزي)

يطلب من :

الدار الدولية للنشر والتوزيع

ص. ب. ٥٥٩٩ هليوبوليس - غرب - القاهرة

ت : ٢٥٨٢٨٨٧

تلکس : ٢٠٠٧٠ PBCRB UN

فاکس : ٢٩١٨٠٥٩ / ٢٠٢٠٢